

Teemu Luomala

# Migration and Enhancement of Plone®-based Multilingual Website to TYPO3® Platform

Helsinki Metropolia University of Applied Sciences  
Bachelor's Thesis  
Industrial Management and Engineering  
August 22, 2011

# **Migration and Enhancement of Plone®-based Multilingual Website to TYPO3® Platform**

## **Bachelor's Thesis**

Author	Teemu Luomala
Degree program	IT Management
Date	August 22, 2011

# Abstract

Author Title Number of Pages Date	Teemu Luomala Migration and Enhancement of Plone®-based Multilingual Website to TYPO3® Platform 52 (including 6 appendices) August 22, 2011
Degrees	Bachelor of Engineering, Metropolia Bachelor of ICT, Hanze University of Groningen
Degree Programmes	Industrial Management and Engineering, Metropolia IT Management, Hanze University Groningen
Instructors	Niels Faber, PhD, Fryske Akademy Tjeerd Rintjema, Fryske Akademy Henk Tattje, Docent, Hanze University Groningen Peter van Kampen, Rijksuniversiteit Groningen Thomas Rohweder, D.Sc. (Econ), Metropolia
<p>The Fryske Akademy hosts the Mercator Research Centre, the Network of Schools, and the Mercator Network websites. They were originally developed on Plone® 'content management system' (CMS) interface which had been decided to be replaced by TYPO3® CMS. Additionally, the accessibility features were inadequate.</p> <p>Little knowledge was available regarding the methodology of converting a Plone-based website to TYPO3 platform. Building up knowledge on this matter as well as clearly documenting the migration were the essential goals of this assignment.</p> <p>All three Mercator websites were successfully migrated during the project. Achieving this positive outcome resulted from a comprehensive analysis of the initial circumstances, and from a determination to design a verified migration method.</p> <p>The templates of all three websites were recreated by committing to W3C's Markup Validation, and accessibility was improved by following the Web Content Accessibility Guidelines. Additionally, the flexible template design allowed straight-forward implementation of "neat features" in the end. The Fryske Akademy went live on the new platform as of finishing the writing of this document.</p> <p>This study teaches that while the actual content migration is merely a mechanical task, recreating existing website's template design under another CMS architecture and/or ideology can be challenging. Nonetheless, redesigning and upgrading an outdated CMS software is an opportunity for an organization to refresh their web presence, and furthermore, renew web editors' toolbox for content management.</p>	
Key words	web content management, TYPO3, Plone, CMS, website migration, website template, TypoScript

## Preface

Thanks to all wonderful people from the Fryske Akademy for providing me this opportunity to put my skills into use and write my thesis in pleasant company during my Erasmus exchange in the Netherlands.

My special thanks to Niels and Tjeerd for generous “right to the point”-guidance.

Thanks to Henk and Peter from Hanze University Groningen for support and especially arranging this project to happen in the first place. This was definitely an educational and interesting learning experience for me.

*Teemu Luomala*  
*Groningen, 22 August 2011*

# Table of Contents

Abstract .....	
Preface .....	
Table of Contents .....	
Abbreviations .....	
1 Introduction .....	1
1.1 Fryske Akademy .....	1
1.2 Project description and objectives .....	3
1.3 Report structure and deliverables .....	4
1.4 Project approach .....	4
2 Analysis of the initial situation .....	5
2.1 The original design .....	5
2.2 Initial content situation .....	8
2.3 Search Engine Optimization .....	10
2.4 Website visitor statistics .....	11
2.5 CMS comparison .....	11
2.6 How to create template for TYPO3 .....	15
2.7 Conclusions of the analysis .....	16
3 The new design and specifications .....	18
3.1 Template design .....	18
3.2 Content specifications .....	20
3.3 Final migration to the live server .....	20
4 Implementation .....	21
4.1 Template engineering .....	21
4.2 CSS style sheets .....	26
4.3 Page structure and content migration .....	26
4.4 Overlapping content .....	27
4.5 Accessibility improvements .....	27
4.6 Search Engine Optimization .....	28
5 Evaluation .....	28
5.1 Markup validation .....	28
5.2 Browser compatibility .....	28
5.3 Content integrity .....	29
6 Conclusions and further study ideas .....	30
6.1 Personal reflections .....	30
References .....	32

## Abbreviations

(W)CMS	(Website) Content Management System (e.g. Plone and TYPO3)
COA	Content Object Array (TYPO3 template design object)
CSS	Cascading Style Sheet
DTD	Document Type Definition
SEO	Search Engine Optimization
SQL	Structured Query Language
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
ZODB	Zope Object Database

# 1 Introduction

This thesis describes the website migration project that took place between May 23rd and August 31st, 2011, at the Fryske Akademy, based in Leeuwarden in the Netherlands. The project has been conducted for the department of social science as the author's graduation project for the School of Computer Science of Hanze University Groningen. The main focus lies on the technical migration and improvements of three individual websites that the Fryske Akademy hosts.

The first section of this chapter gives a brief introduction to the Fryske Akademy. In the second section, the project problem is introduced, as well as the goals set in the beginning of the project. The third section describes the structure of the rest of the document, and description of project approach concludes this chapter.

## 1.1 Fryske Akademy

The Fryske Akademy is the Frisian research institute, founded in 1938, carrying out scientific research in the fields of the Frisian language, culture, history and society. They co-operate with various academic institutions, both in the Netherlands and abroad by organizing lectures, conferences and symposiums for instance. They take part and host some EU-projects in the field of minority languages, as Frisian is one of the endangered languages and cultures in Europe.

A great amount of the Fryske Akademy's work focuses especially on the West Frisian language; one of their major publications is the "Wurdboek fan de Fryske taal" (Dictionary of the Frisian language), which stands as the language's principal authority. The Fryske Akademy's long-term main objective quoted from their website is: "Maintaining a study group that focuses on research concerning Friesland (Fryslân in West Frisian), the Frisian people and the Frisian culture, all in the broadest sense."

### *Funding and organization structure*

The main operation of the Fryske Akademy is made possible by funding of the Royal Netherlands Academy of Arts and Sciences (KNAW), the province of Fryslân, and the municipality of Leeuwarden. The daily overall management is run by a managing director, who receives assistance from a Management Team, the Supervisory Board, the Science Committee, and the Social Advisory Council. The organization has around 60 employees. Additionally about 300 scientists, professionals and amateurs are active in the various scientific societies that are hosted by the Fryske Akademy. The Fryske Akademy has three main departments for its activities, namely: history, linguistics, and social science.

### *Social science department*

The management of the websites that are migrated in this project belongs to the social science department. The following table presents the people from the Fryske Akademy who were significantly involved in this project.

*Project supervisors, (technical) design and documentation feedback*

Dr. ir. N.R. Faber (researcher) and T. Rintjema (web editor)

*Access to the IT systems and help with the network configurations*

D. Drukker (IT developer)

*Final acceptance of the deliverables*

Drs. C. van der Meer (Leader of the Mercator project)

Prof. Dr. R.J. Jorna (Head of Social Science department)

Prof. Dr. R. Salverda (Managing director)

The websites that are migrated in this project are public part of major projects and thus they have their individual websites rather than being under the Fryske Akademy domain. A short description of each project is given in the following subsections.

#### *The Mercator European Research Centre on Multilingualism and Language Learning (Mercator Research Centre in short)*

The Mercator Research Centre is the leading partner of the Mercator Network that consists of five research and documentation centers. Each of the centers has its own thematic program and specialist role related to minority and regional languages within the European Union. The Mercator European Research Centre on Multilingualism and Language Learning receives funding from the province of Fryslân and the municipality of Leeuwarden.

A mission statement citation from their website: "In the 21st century all European citizens live in a multilingual environment. In their daily lives they come across many different languages. Consequently there is a growing interest in multilingualism and an increasing need for language communities to exchange experiences and to cooperate. The Mercator European Research Centre on Multilingualism and Language Learning, part of the Fryske Akademy, meets this need by collecting, researching and studying, circulating, and applying knowledge in the field of language learning and use at school, at home and through cultural participation."

The goal of the Mercator-project is summarized in the following quote from the website: "The Mercator Research Centre is an independent and recognised reference centre for policy makers and professional workers in the field of multilingual education and language learning. The Mercator Research Centre brings Europe to Fryslân and Fryslân to Europe."

The website content includes plentiful research publication content such as regional dossiers, and other country-specific detailed information about minority and endangered languages around Europe as well as various reports and studies categorized under individual project-pages. There is also a frontend to a library with more than 7300 entries, databases of more than 450 linguistics experts and 1200 organizations, frequently updated news content, language conference information, as well as extensive external link-archive for instance.

#### *The Mercator Network of Schools*

The Network of Schools is Fryske Akademy's digital platform for multilingual schools of minority language areas of Europe for exchanging information among educational



practitioners. It is a subproject of the Mercator Research Centre. The website has been created and used in order to encourage contact between the different member schools and their pupils, and promote collaboration on joint projects so that they learn from each other directly.

The website content includes for instance a clear presentation of all the schools that are part of the still expanding network, frequent news bulletins, information regarding funding possibilities, overviews of publishers who issue teaching materials in minority languages, project reports, and plentiful amount of links to related websites and other publications.

#### *The Mercator European Network of Language Diversity Centres*

The Mercator Network is an EU-funded project between five partners of which the Fryske Akademy's Mercator Research Centre is the leading one. This Network builds on the achievements of the former Mercator Network, which was founded in 1987. The objective is to connect multilingual communities across Europe, promote knowledge sharing and facilitate structured exchanging of best practices and cutting-edge initiatives. The main focus lies on multilingual regions dealing with regional or minority languages, but also immigrant languages and smaller state languages, with an emphasis on language needs arising from migration and globalization.

This website is not as content rich as the other two as it merely serves as a main portal for the project activities and its main members' websites, but frequently updated conference, expert meeting, and various news pages, make it an independent website on its own regardless.

## 1.2 Project description and objectives

Prior to this project the Fryske Akademy had made a decision to have all their websites modernized. In the beginning of this project in May, their main website [www.fryske-akademy.nl](http://www.fryske-akademy.nl) had just been upgraded by a commercial company and the upgrade was undocumented. TYPO3 had been chosen as the new website content management system (CMS) in consultation with the Fryske Akademy.

*The practical main objective of this project is to migrate three individual websites that were introduced in the previous chapter, namely: the Mercator Research Centre, the Mercator Network of Schools, and optionally, the Mercator Network, so that they would be running under the same content management system installation as the main website.*

The existing CMS platform, Plone, for the Mercator websites was chosen to be abandoned because of maintainability and poor upgradeability reasons. The current situation of running two separate CMS applications required time consuming extra maintenance and management, so it had been decided as a desirable future scenario to have all websites running under one TYPO3 installation.

Along the same line of reasoning, a realization was made that this is an opportunity to modernize the websites, not to just clone the sites as they are to another system.

It was going to be necessary to not only learn how to use TYPO3, but also realize how it functions technically in order to actualize everything needed to recreate the websites.

This thesis 1) *describes how the website migration from Plone to TYPO3 platform was implemented*, as that was the main objective of this project next to 2) *the actual migration of the websites*. The third objective was to 3) *provide recommendations to improve the accessibility and execute them as far as the time permitted*.

### 1.3 Report structure and deliverables

The project problem is analyzed in detail in chapter two. The following chapter describes in detail what design decisions and specifications were made and why. Chapter four presents the description of the practical implementation of the migration, and chapter five discusses evaluation of the deliverables. The sixth chapter concludes the report with final discussion.

#### *Appendices*

Because some custom design solutions have been made in order to achieve the desired results, some manuals have been written on specific technical matters for the Fryske Academy employees that work on managing the websites. The appendices are provided only to the persons on the distribution list of this document.

#### *Deliverables*

The final deliverables of the project were defined as presented in the following list:

1. Documentation describing the migration process of a Plone-based website to TYPO3 platform
2. 'Mercator Research Centre' and 'Mercator Network of Schools'-websites
3. An extra objective: 'Mercator Network'-website

### 1.4 Project approach

The initial challenges to study are about standard-compliant website design definitions, and how Plone and TYPO3 applications function technically in order to be able to replicate the existing websites' design and perform the content migration. In order to improve the accessibility of the websites, research on currently valid and legitimate guidelines must be done. As the websites' content has to remain intact, a method has to be found that would allow the content data to be copied with the lowest probability of error, while leaving room for the accessibility and other improvements to be done.

#### *Time schedule*

The end of August was set as a deadline for the project. A following tentative schedule was planned and it shows the intended progress of the project:

<i>Analysis</i>	
Week 1 (23.5-)	Analysis of existing websites and learning to use TYPO3

Week 2 (30.5-)	Analysis of existing websites and learning to use TYPO3
Week 3 (6.6-)	Learning to use TYPO3 and research about templates
<i>Design</i>	
Week 4 (13.6-)	Template design and implementation and start content migration
<i>Implementation and testing</i>	
Week 5 (20.6-)	Content migration – Goal to finish migrating ‘Network of Schools’
Week 6 (27.6-)	Content migration
Week 7 (4.7-)	Content migration
Week 8 (11.7-)	Content migration – Goal to finish migrating ‘Mercator Research’ if possible
Week 9 (18.7-)	Content migration and writing the report
<i>Reporting and evaluation of the final deliverables</i>	
Week 10 (25.7-)	Goal to finish the migration
August 22	Deadline to finish writing the report.

## 2 Analysis of the initial situation

In order to determine *the most sensible solution to perform this specific migration*, it is necessary to study and understand a) basics of (technical) website design, b) the existing websites’ design and content, c) technical functionality of both content management systems in question, and d) basic logic and purpose of the CMS-based website administration.

### *Why use a CMS*

A basic purpose of using CMS is to make the management of the website content simpler to handle, and from a technical point of view: separate the design and content more explicitly. If the website design (template) is set up properly, the business can put their focus on the actual content management. (Battles, 2008)

### 2.1 The original design

#### *What is a template*

A carefully designed template is the very basis for a successful website to be operated under a content management system. Template is the skeleton in which the dynamic content is inserted to from the content management system’s database at every page load. All static content that is displayed on every page of the site, and the objects that give the website its visual appearance, are defined in the site’s template design. In this way the layout will be consistent throughout the whole website. (Skårhøj, 2003a)

#### *The original template*

The initial finding was that all three original Mercator websites share the same basic template design; the only major differences being a different color theme, and the ‘Mercator Research Centre’ is the only fully multilingual site.

The websites were based on a template that comes with the CMS. The installed version of Plone at Fryske Akademy was 2.5.5 from 2007. This was one of the main reasons for websites' somewhat outdated "Plone look".

The following two images (figures 1 and 2) give a clear demonstration of the websites' changing behavior due to the fact that their CSS styling settings allowed the main content container to stretch to whatever horizontal resolution available. Moreover, the second image (figure 2) demonstrates the four main container elements of the original design, and the following text provides a more technical analysis based on it.

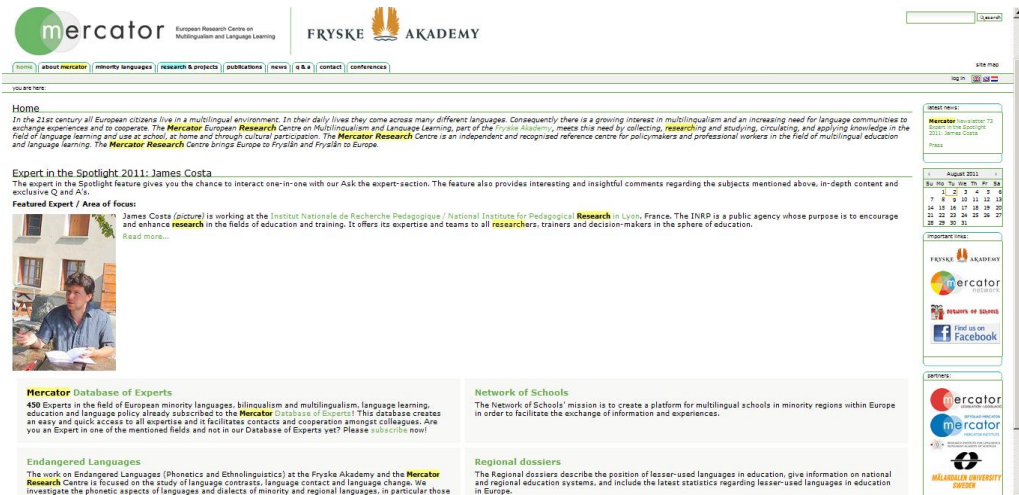


Figure 1. A widescreen resolution screenshot of the original website design.

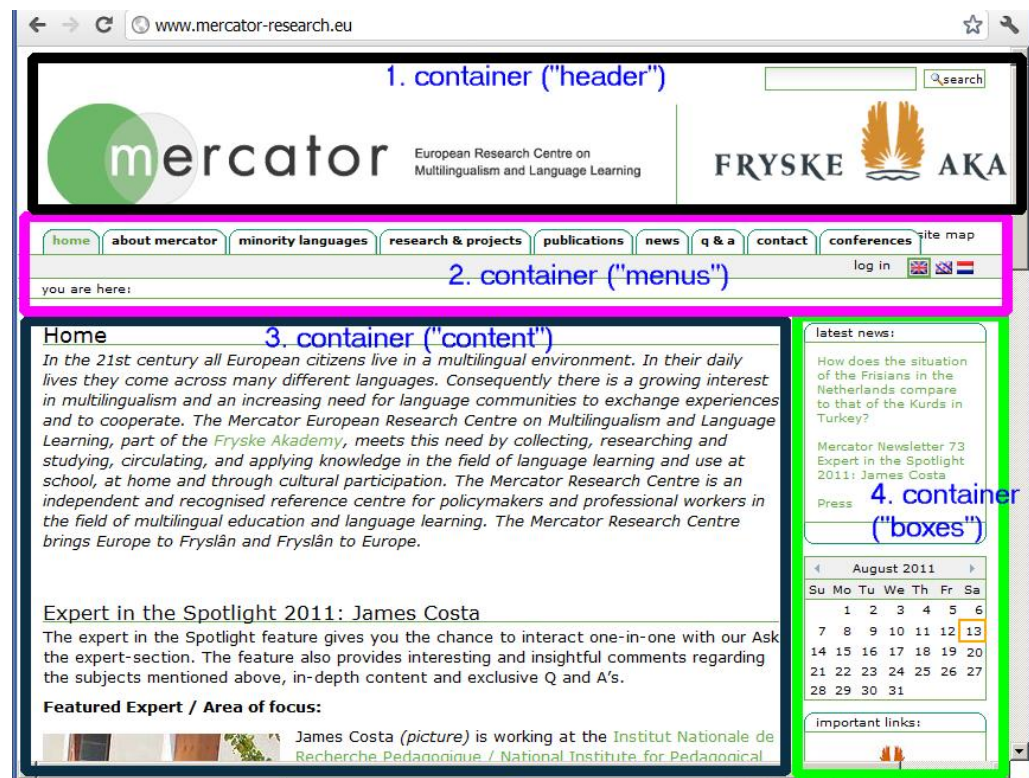


Figure 2. Low resolution screenshot of the original design + four main container elements highlighted.

1. The first main-container contains 1a) logo images, and 1b) a search function.

2. The second contains 2a) a navigational menu, 2b) a separate site map link, 2c) three language-switch buttons, 2d) a “log in” link for administration of the site (which can confuse the regular site visitors), and finally 2e) a breadcrumb trail (‘you are here’ text), which should show the currently open page’s title and path. In this case the homepage of the website is open and thus there is no page path to display because homepage is (usually) the root page.
  3. The third contains 3a) the main page content.
  4. The fourth contains four content “boxes” which are so big that they all don’t fit to be displayed at once. The first “box” shows 4a) the latest news as text-links, 4b) internal Plone calendar, 4c) important links as images, and finally, 4d) partners (as image links).
  5. At the very bottom of the page, there was also a footer element, but it didn’t contain anything but a “Plone Powered” image link.
- “Network of Schools” and “Mercator Network” had exactly the same design elements, except they did not have the language switch buttons (2c) or calendar (4b).

### *Static versus dynamic content*

The previous section introduced all the building block elements of which the original websites consist. Now it is important to determine which of the content elements are static (always the same throughout the website), and which are dynamic (updated based on which page of the website is currently open). This makes a fundamental distinction between the static design and actual page specific content, and ultimately what is important to this project, the way they need to be migrated.

1. The first container, including all the elements in it, is completely static.
2. Inside the second, except for “site map” link (2b) and “log in” (2d) link, the contents change dynamically:
  - a. 2a) Menu stays the same content wise, but the active item (the page that is open; in this case the “home”-page) of the menu gets highlighted with grey background. This feature was found to be implemented as a CSS style definition which technically requires specific ID for the active menu item in order to be distinguished from the other menu items.
  - b. 2c) The same as above goes for the language switch buttons, except the active language is highlighted with a green border.
  - c. 2e) The breadcrumb trail shows the current page title (and path if it is a subpage), so it changes dynamically.
3. The main content obviously changes dynamically.
4. The “boxes” elements were found to be *site wide* elements, meaning that their design or content did not change no matter which page of the website was open.

### *Resemblance to the main website and multilingualism*

The upgraded Fryske Akademy website’s design choices are also analyzed in order to achieve some requested resemblance to it with the Mercator websites. The initial design resemblance features to be applied are 1) a positioning of the language switch buttons on the left side, 2) a font resizing accessibility feature, 3) the calendar, and 4) strict horizontal width and center aligning (in CSS) instead of allowing the main content container to stretch to whatever width. This stretching effect was specifically requested to be fixed by the web editor.

The Fryske Akademy website was also found to be successfully configured with multiple languages under TYPO3. TYPO3 has an internal support for creation of multilingual websites. This meant that there were no major challenges identified regarding that aspect. The existing method of implementing the multiple languages from the main website was found to be most sensible to replicate in order to keep the consistency, and to avoid different implementations of the same configuration.

#### *Original CSS style sheets*

The CSS styling of the original websites was divided into 19 separate CSS-files for each site, with some settings being defined multiple times. This makes it time consuming to determine which settings are ultimately responsible for the final look of the websites. Thus the reduction of the number of separate CSS files and redundant settings is identified as an important task to accomplish.

#### *HTML output*

The HTML code output of the original website pages was analyzed. It was found to be technically complex and it produced a lot of errors in the Markup Validation Service test (W3C, 2011). This is not going to be acceptable after the migration.

#### *Original template analysis conclusion*

The template design is found to be realistic to recreate under TYPO3, but redesigning is required anyhow to make the websites more accessible and simpler to manage.

## 2.2 Initial content situation

Before delving into the technical functionality of CMS applications, the content output of each website is examined.

#### *Mercator Research Centre*

The Mercator Research Centre (figure 1 on page 6 presents the front page) is identified as the most complex and content rich website out of the three. Not only is it a multilingual website, it also has the highest amount of content-pages (200); including a) Dutch and Frisian translations of most of the pages, b) special pages such as a calendar events and forms, c) 250 image files, and d) 210 downloadable files.

#### *Network of Schools*

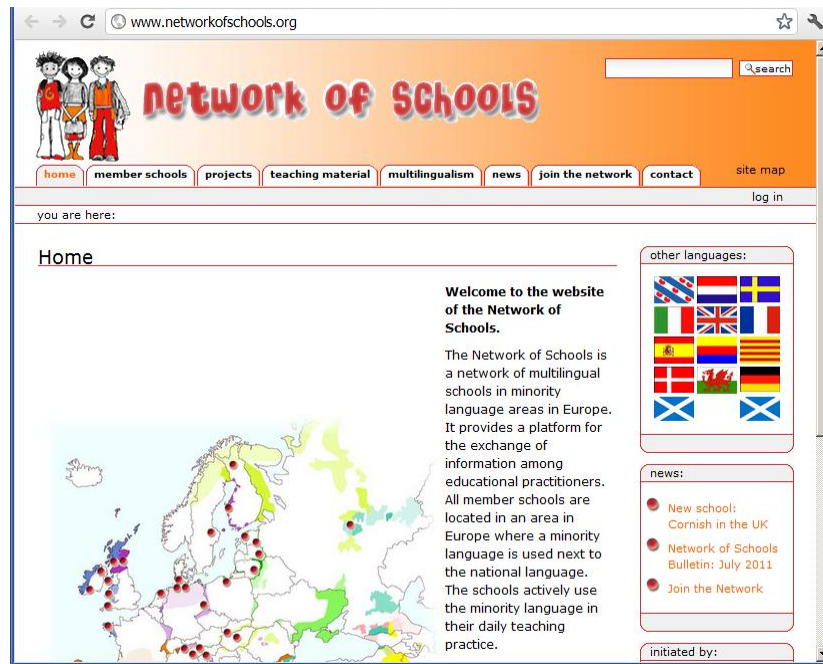


Figure 3. The original layout of the 'Network of Schools'-website.

The content of the 'Network of Schools' website appears simple at the first glance, but some of the pages contain many links and images. A summary of the public content is 75 individual pages, 215 image files, and 19 downloadable files.

#### Mercator Network



Figure 4. The original layout of Mercator Network –website.

The Mercator Network website is found to contain none challenging pages to migrate. A summary of the content is 40 individual content pages, 75 image files, and 20 downloadable files.



### *The changing content -problem*

The original websites are going to continue being updated and maintained as normal during the implementation. However, there are aspects that are going to decrease the difficulty level of this particular migration: 1) a quiet time of the year, 2) none of the sites are updated more frequently than few times a week, and hence 3) the majority of the content can be considered static. These conditions allow low priority on keeping up to date with the changes to the websites during the implementation as it is quick to find out if anything is changed from the change logs of the CMS.

### *Image files*

Some of the images were found to be missing <alt> tags, or they were uninformative. Web Content Accessibility Guidelines (WCAG) state that “a text equivalent for every non-text element shall be provided” (W3C, 2011). This is because there are e.g.: a) text based web browsers, b) for many possible reasons the image display can be disabled or image file is unavailable, and c) blind people also browse the Internet by having the information on the web pages spoken instead. This is the main reason why at least the informal images should have text equivalent defined.

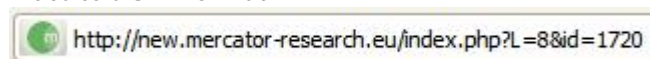
### *File directories*

The original web server’s file directory was found to be in a need of maintenance. Some of the image files and downloadable files were misplaced in illogical directories. It is identified as an important aspect to be improved when copying the files to the new server. Not only because all files would then be organized, but the directory names and especially the filenames of images and downloadable files appear in the public HTML output of the website regardless. This aspect can have relevance if SEO (Search Engine Optimization) is considered important. The following section gives more insight to this aspect.

## 2.3 Search Engine Optimization

It was requested to improve the findability of the websites. The original websites had none meta keywords defined. This could be easily fixed by allowing them to be defined in the new template.

The original websites had speaking URLs, which means that the URL (website address) to a page on a website consists of meaningful word(s) (usually the title of the page) instead of meaningless variables such as the default TYPO3-based website’s URL-format:



Where the “index.php” is always there, the changing parts being the “L=”-attribute for an active language, and numerical “ID”-attribute for a currently shown page. Thus the default output of the public URLs from TYPO3 is not word search friendly, so it was decided that solution should be found that would translate the default URLs.

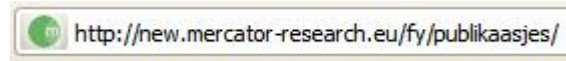
### *RealURL*

There were multiple extensions found that translate the ID based URLs to speaking kind, but the most stable (programming bug free) available software to do the job was found out to be an extension called RealURL. The default “index.php”-based



URLs also still function as normal even after the extension is activated, so there is no problem of linkrot when using the RealURL. Linkrot is an informal term also known as “broken link”, which means an URL that is not available anymore, but a link to it still exists somewhere (Gomes & Silva, 2006).

After a successful installation and configuration of the RealURL extension, the above URL would look like this:



The “?L=8”-language string is now displayed as “fy” which is the defined two word abbreviation for Frisian language. The TYPO3 page-ID string “?id=1720” is translated to the real title of the page instead, “publikaasjes” in this case. This not only gives more professional appearance to the visitors of the site, but the links to the speaking URL-enabled site look more sensible. This is also SEO-friendly because the keywords are typically highlighted in the search results, and nevertheless, the words in the URLs are taken into account as keywords for the ranking of the website. (Coles, 2009)

## 2.4 Website visitor statistics

Studying the internal Mercator-websites’ visitor-statistics provided by the Fryske Akademy (no public access) revealed that the users with lower horizontal resolutions than 1024 pixels are practically so rare, that it would be safe to choose a width of 1024 pixels as the lowest resolution in which the website must display without having to scroll it horizontally. The Fryske Akademy website statistics had likely given similar results already earlier, because 988 pixels had been chosen as that website’s strict horizontal width.

Schools, libraries and universities were told to form a significant portion of visitors to the Fryske Akademy and Mercator websites. The default web browser at the mentioned locations is often (even restricted to) Microsoft Internet Explorer, which (especially the older versions than 7) often has problems displaying even standard compliant websites correctly. As of writing this report, for example version 6 of Internet Explorer is still being used a lot worldwide as Microsoft’s own research has revealed (Microsoft, 2011). Internet Explorer is regardless going to get extra attention in the testing later.

## 2.5 CMS comparison

Now that the design and contents have been explored, the method to transfer the content-data still remains to be discovered.

The basic logic how the different CMS applications function and save the data differs. Or even if the logic sometimes is similar, at least the names and values of objects and tables for the same things hardly ever match. Thus the database-data hardly ever is directly transferable between “competing” applications, open source or not. In case of Plone, even internal migrations between different versions of Plone were found to be tricky (Quintagroup, 2010).

A full administrator access to the existing CMS-platforms and databases were given, which allowed deeper technical analysis of them.

### Web editor's comments

A brief interview with the Mercator-sites' web editor was held to get case specific insight into differences of Plone and TYPO3. These were his main thoughts about the change:

"TYPO3 does what you expect it to do. Also it generates better HTML code. I also like the option to see the three different languages side by side. It provides a better overview, but this is also inherent to the backend editing interface."

"Images are easier to place in the right position. In Plone the vertical and horizontal spacing option didn't function properly, I always had to change the HTML code manually."

"Creating new pages is easier in TYPO3, especially the second and third language versions. In Plone it was cumbersome to get all three languages similar and approachable in all different ways. For instance it was easy to create a situation in which the Dutch version of a page was available if you clicked on the Dutch flag while in that particular page in another language, but not directly from the Dutch pull down menu. In TYPO3 these kinds of mistakes seem impossible to make."

### Plone

The Plone CMS is programmed on top of the Python based Zope (Z Object Publishing Environment). By default all data is stored in Zope's built in transactional object-oriented database (ZODB in short), and this is the case here. As all objects are defined in Python code, a corresponding Python source code needs to be installed in order to instantiate the objects from ZODB.

From external file-directory approach the database is (usually very large) single file called a FileStorage (Plone, 2011). Inside Zope interface all database objects can be seen as "binary large objects" (BLOBs). It was found that all BLOBs can easily be accessed inside the installed Zope interface as the following image (figure 5) demonstrates.

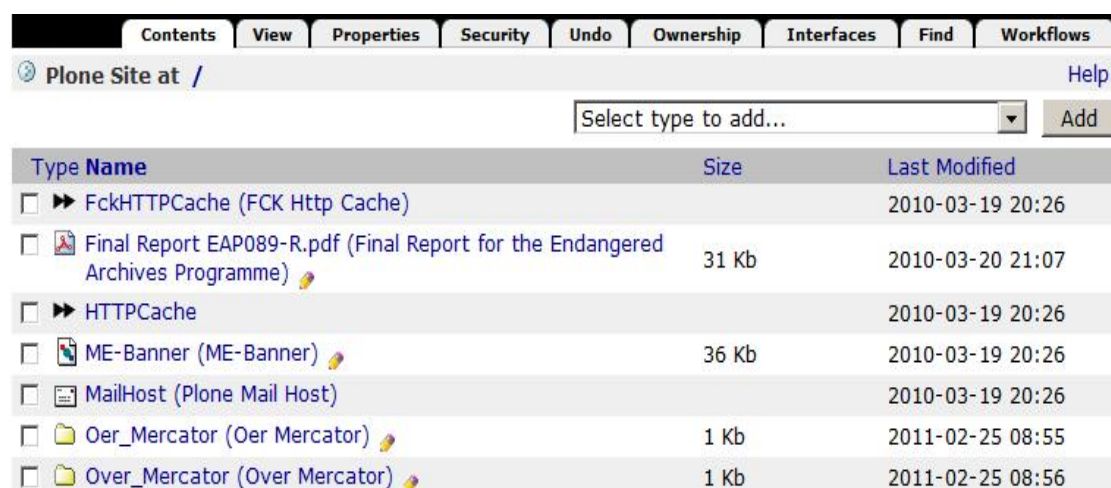


Figure 5. Zope-interface's content-view which shows all objects inside ZODB.

In Plone interface, instead of seeing all objects, only the webpage content related objects are visible (figure 6).

[home](#)
[about mercator](#)
[minority languages](#)
[research & projects](#)
[publications](#)
[news](#)
[q & a](#)
[contact](#)
[partners](#)
[conferences](#)
[downloads](#)
[site map](#)
[site setup](#)

[admin](#)
[preferences](#)
[undo](#)
[images](#)
[log out](#)

you are here:

[contents](#)
[view](#)
[edit](#)
[sharing](#)
[syndication](#)

display ▼ add item ▼

**Mercator**

<input type="checkbox"/>	title	size	modified	state	order
<input type="checkbox"/>	Home	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Images	1 kB	2011-06-07 16:07	Published	▲ ▼
<input type="checkbox"/>	about Mercator	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Minority languages	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Research & Projects	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Publications	1 kB	2011-06-23 13:09	Published	▲ ▼
<input type="checkbox"/>	News	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Q & A	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Contact	1 kB	2010-03-20 20:40	Published	▲ ▼
<input type="checkbox"/>	Partners	1 kB	2011-03-11 11:04	Private	▲ ▼
<input type="checkbox"/>	conferences	1 kB	2011-03-18 08:12	Published	▲ ▼
<input type="checkbox"/>	Important links:	1.2 kB	2011-07-28 16:31	Published	▲ ▼
<input type="checkbox"/>	robots.txt	1 kB	2006-09-20 19:27	Published	▲ ▼
<input type="checkbox"/>	Read about Mercator in 23 languages	4.7 kB	2006-10-30 16:22	Published	▲ ▼
<input type="checkbox"/>	stats	1 kB	2006-10-16 11:53	Published	▲ ▼

latest news:

How does the situation of the Frisians in the Netherlands compare to that of the Kurds in Turkey?

Mercator Newsletter 73 Expert in the Spotlight 2011: James Costa

Press

August 2011

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
7	8	9	10	11	12	1
14	15	16	17	18	19	2
21	22	23	24	25	26	2
28	29	30	31			

important links:

Figure 6. Content-view of Plone CMS interface.

From Plone's content only perspective objects were logically arranged, however in Zope's 'full database view' objects could have been better organized. This is the advantage (and drawback) of an object oriented database where one has more flexibility in organizing and manipulating the data without losing the objects' relevancies to each other.

If the functionality and structure are well studied and realized, ZODB type of database can flexibly be customized according to circumstances to make management of the websites less time consuming to handle. In case of a less technically oriented people the potential of the database usability is often missed, even misused. (Gift & Rhodes, 2008)

### TYPO3

TYPO3 is PHP-based and it supports many relational databases. However, MySQL is the database already used by the Fryske Akademy with the main website, and it is more sensible to use the applications that are already configured and operational.

TYPO3 is an independent application meaning it is not built on top of any other software as Plone is. TYPO3 has only one main administrative backend interface in which the content management takes place. Its logical database structure also makes it easier to approach for non-technical persons because the pages of the website must be logically structured under the page tree.

The following image (figure 7) presents the backend interface display where the page tree structure is visible in the second column from the left.

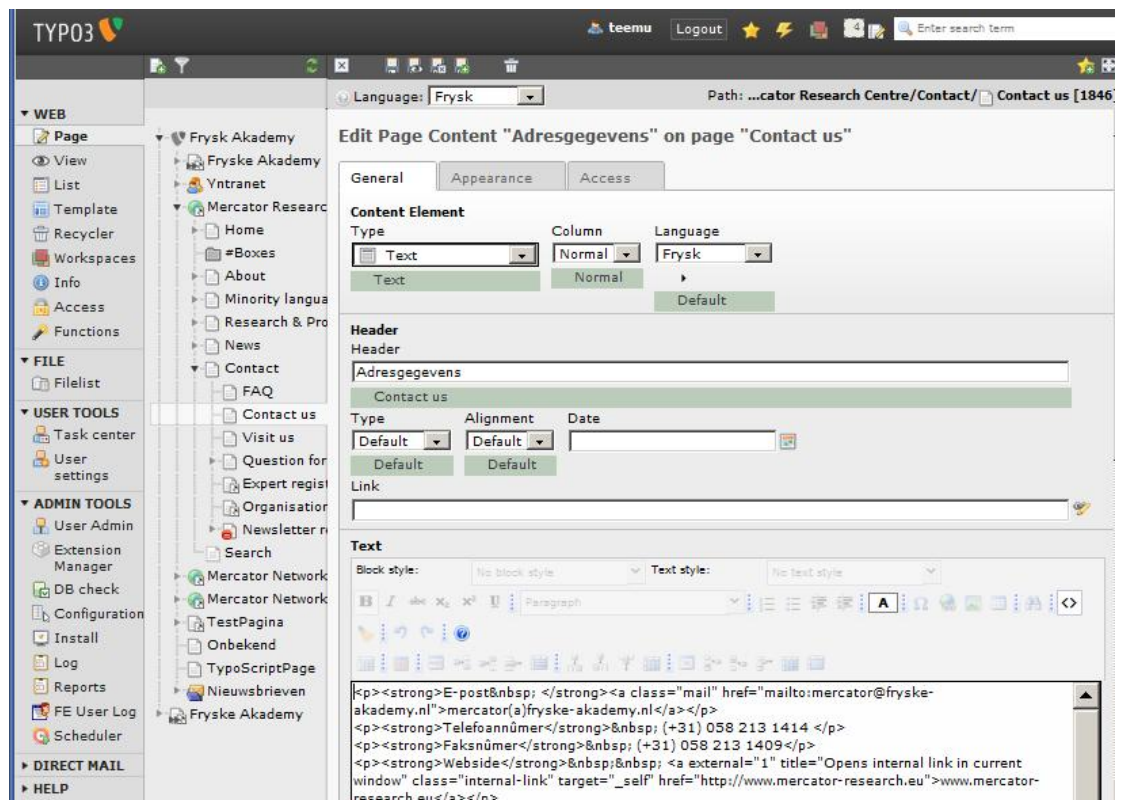


Figure 7. TYPO3 backend interface. Shows page tree structure and default page editor.

In the above image, TYPO3's default webpage editor is also shown in HTML mode. This is similar to Plone's equivalent as the following image (figure 8) shows.

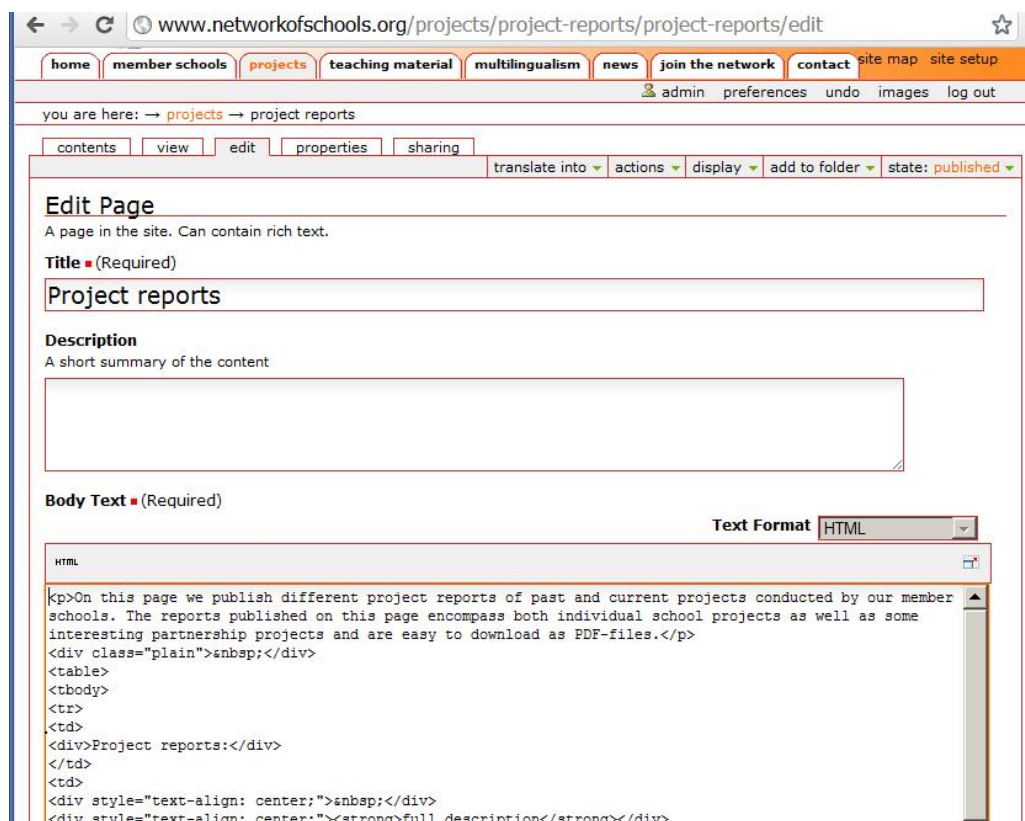


Figure 8. Default webpage editor of Plone.

Here a realization was made that the page content from the above HTML-view can be directly copied and inserted into TYPO3's equivalent.

#### *Testing the discovered method*

The first migration using the above method was performed in a local fresh installation of TYPO3. The content pages of "Network of Schools" website were directly copied into TYPO3 page by page from HTML view of Plone's page editor. After the copying, all the internal links (links to other pages within the site) had to be fixed manually. Links to external sites did not require fixing.

The image below shows the syntax of internal Plone-links and images.

```
<a href="/resolveUId/980c5253eefc967ba2c25d14ffcaef7b"> tag, it was not accepted by TYPO3's page editor by default, but two workarounds were found: 1) The settings of the editor can be changed, or 2) whatever HTML content can be inserted; by not creating a page but a special HTML content element instead, which is available in TYPO3 by default.

All content pages of "Network of Schools" website were successfully migrated using the above method. As it was found out that internal links and images had to be fixed manually anyway, and in order to improve the accessibility of the image files (by changing the filenames and by adding text descriptions), this migration method was concluded to be the most promising with the lowest probability for error.

#### *Conclusion of the CMS comparison*

Given the established deadline for the project, there is no time to delve into figuring out how ZODB-data could be extracted using a script-method; neither is this migration project going to require it. Thus it is calculated to be most sensible to invest time to *create a new template from scratch* that would not only 1) preserve the basic functionality and basic visual design of the original websites, but 2) enhance the accessibility, and 3) allow other planned improvements to be simple to implement, and ultimately, because 4) TYPO3 is the software the Fryske Akademy is aiming to be using after this project solely.

## 2.6 How to create template for TYPO3

Creating a standard compliant template to be used with TYPO3 requires not only visual website design skills, but a technical understanding of HTML, CSS styling, and some understanding of TypoScript is also necessary. A comprehensive guide how to create a TYPO3 template is found on the official TYPO3 website (Skårhøj, 2003a).



### *HTML-template*

The website design process usually begins with a graphical vision of the website that is to be created. In this project, the visual designs already existed, but they were being re-created from scratch and if possible, improved in the process.

As was explored in figure 2 (page 6), it is already known what sort of building blocks the original websites consist of. This is what is needed to be known in order to design and implement a new HTML template.

### *What is TypoScript*

TypoScript is a TYPO3-specific configuration script language basically to instruct the TYPO3 what to do exactly when the web browser application requests a page to be displayed from TYPO3-based website.

TypoScript is basically a medium between the internal TYPO3 functionality and the external website visitor's browser application. When a TYPO3-based website is visited, it works as any website logic: a) the visitor's browser requests a certain page to be displayed, b) the TYPO3 website responds and based on the configuration instructions defined in the TypoScript code, and c) TYPO3 outputs the content as defined in the building blocks of the HTML template. Finally, the final visual appearance of the website is adjusted in the CSS style sheet(s).

### *TemplaVoila!*

A more recently published template guide (Skårhøj, 2004) suggests creating TYPO3 websites using TemplaVoila! extension. TemplaVoila! makes creation of templates more visual, but it also changes how TYPO3 works by default in many ways, and none of this is desirable or needed in this case, because it would lead to non-standard functionality and a more time consuming administration.

Ultimately, it was also found to cause a technical conflict with the already migrated Fryske Akademy website which does not utilize TemplaVoila!, and it is never sensible to modify the existing working configurations unnecessarily.

## 2.7 Conclusions of the analysis

As the design and content of the websites in CMS-environment are (supposed to be) separate, it is concluded that their migration has to be performed separately as well.

### *Design issues*

A conclusion to the screen resolution issue is that if there is enough time and it is considered worth the effort, a separate style design can be created for different resolutions, and e.g. for the users of mobile devices.

A conclusion to the web browser compatibility issue is that the new template will be created by following standards, as then they should function flawlessly with the most browsers used worldwide. However, a manual testing is required to witness how specific website functions in specific browser versions. If a non-working browser version is found and it is wanted to be fixed, there always is some sort of

workaround. If nothing else is possible, a totally separate CSS design for certain browser versions can be created next to the default design.

The 'Mercator Research Centre' website will require additional design and engineering to implement the support for multiple languages. The existing method from the Fryske Akademy website appears to be the most sensible to replicate.

#### *Content issues*

As 1) the way CMS-application stores the values in the database are platform specific, 2) no existing methods were found how to export and import content data automatically from Plone to TYPO3 directly, and 3) even a programming of a script for automatic exportation of the content data directly from the ZODB database was found to be possible, 4) but given the established deadline and the little potential for reuse of that sort of script; the script programming method is concluded to be unpractical and too time consuming work to do in this case.

Another central observation is that the amount of actual content on the three websites is not that massive that it couldn't be migrated manually page by page in the available time.

Performing the content migration manually page by page also has the advantages that 1) all steps are traceable, 2) the integrity of the migrated content is instantly verified, 3) the improved file naming, 4) accessibility, and 5) search engine optimization features are instantly applied as well.

## 3 The new design and specifications

### *Web Content Accessibility Guidelines (WCAG) compliancy*

The template design is going to be completely based on a standard compliant XHTML and CSS style sheets which should validate when submitted to W3C's Markup Validation Service (W3C, 2011).

This means that that one central principle is the avoidance of JavaScript because 1) none of the truly accessible website's basic functionality should ever rely on JavaScript, 2) JavaScript is not universally supported, and even if it is, 3) some public locations have restricted or even disabled it due to security reasons, and ultimately, 4) the original sites did not have any JavaScript functions for the basic design.

Another decision is that all three sites would have to share the same basis template and main CSS design settings, 1) to simplify the management, and 2) the sites were similar in design originally and this would not only 3) make the websites more interconnected, but would also 4) help the visitors to recognize the websites as belonging to the main entity, the Fryske Akademy.

### 3.1 Template design

As figure 2 (page 6) demonstrated, there were clearly distinguishable content containers identified that formed the basic design. This is all that is needed to be known in order to get started in recreating a new template from scratch. Flexible future improvement possibilities over the websites' outdated look are taken as an extra objective next to the accessibility improvements.

The following image demonstrates the improved layout visually that was designed in consultation with Fryske Akademy. This 3 column design does not exactly parallel, but at least follows the ideology behind the so called "Holy Grail" of the CSS-layouts, which translates to 1) no CSS hacks, 2) SEO friendly, 3) no JavaScript, and 4) resizable text compatible. (Taylor, 2007)





Figure 9. The new template design.

1. The first container now has background gradient (the original site did not have it), and includes 1a) two logo pictures (Fryske Akademy logo is separated and improved), 1b) "site map" link (new position), 1c) "search-box" element (as original), 1d) two 'font-resize' buttons (new accessibility feature to be implemented in JavaScript), and finally 1e) the main navigational menu (new positioning and accessibility is improved with larger font size). They are all individual elements that are going to be positioned to their specified positions in the CSS style sheet.
2. The second container includes 2a) three language switching buttons (resemblance to the Fryske Akademy website), and 2b) a breadcrumb trail (as original).
3. This container is a new element compared to the original design. By dividing the 'content boxes' to both sides, this not only a) makes the list shorter, but b) allows presentation of some important content to top of the page on both sides, and c) the main content element (4) is now in the center of the layout. Instead of finding out a way to show the calendar as it was on the original site, the new design has a content box instead with direct links to upcoming event(s).
4. This is where the main content of the pages goes.
5. Same as the third container, except content is from different source.
6. A static copyright disclaimer of the website owner is placed in the footer.

As mentioned in the first chapter, the websites in this project have a lot in common. The basic template layout that is demonstrated in the figure 9 is going to be the same for each site. This choice in the end results to a simpler website maintenance.

To preserve each site's unique visual appearance, the original color theme of each site is going to be preserved. The color data is found from the old websites' site-specific CSS-style sheet settings.

### 3.2 Content specifications

The front page of the Mercator Research Centre was requested to be improved with something innovative. The original front page design was created as outdated looking table structure, in which four little content "boxes" were present. An extension called "Multiple Content" was approved to be installed in order to show multiple page content items at once more innovatively.

The special content pages; calendar events and forms, are specified to be explicitly manually re-created inside TYPO3, because no method was found to automate their migration.

As the 'Network of Schools' had already been migrated successfully during the testing phase, there are no further specifications for the content copy implementation.

### 3.3 Final migration to the live server

The websites are initially migrated to the development server from where they will be eventually copied to the live server where the main Fryske Academy website already is running on. This internal TYPO3 content migration had also been tested during the initial test migration of 'Network of Schools' website, so there are no known issues to be concerned about.

The upgrading of the network configurations for the new sites is going to be IT department's task in the end. Moreover, when the development server has been set up, the new sites can already be given new temporary public URLs which allows testing of the websites from the public Internet.

## 4 Implementation

The IT department had set up a full clone of the live TYPO3 installation where the main website is running on. This allows working in the exact copy of the environment where the other websites are going to be migrated into.

The initial template design was already partly created during the testing phase. This initial template is now extended to include all the elements and functionalities as designed. Same goes with the content migration, as one site had already been migrated, the rest are going to be migrated in the same way. In the end, the focus shifted into finalizing the websites with visual design improvements based on feedback, and implementing some “fancy” features.

### 4.1 Template engineering

To understand how it all comes together in practice, the HTML template is examined first because it is the primary template element where all the key elements are first defined, and ultimately, where the final result is inserted into.

#### *HTML-template*

As was designed in figure 9 (page 19), it is known how the new sites were desired to be like. These basic website building blocks are now defined in the HTML. The following code snippet is a simplified version of the actual HTML template. All rows are given explanation below.

*Note: The code presented below is not fully functional HTML, it is purposefully missing e.g. the closing tags of some of the elements, image links, and other irrelevancies to this report, because the objective here is to explain functionality of TYPO3, not basic HTML. The complete code listings are included in the appendices of this documentation.*

```
[1] <div id="header">
[2] <div id="fontbigger" class="ifNoJS"><a onclick="resizeFont(1);">... </div>
[3] <div id="fontsmaller" class="ifNoJS"><a onclick="resizeFont(-1);">... </div>
[4] <div id="main_menu"><!--###main_menu###-->Main menu comes here<!--###main_menu#...</div>
[5] <div id="lang_loc">
[6] <div id="flagfy"><!--###FLAGFY###-->Frysk flag comes here<!--###FLAGFY###-...</div>
[7] <div id="container">
[8] <div id="boxes_left"><!--###boxes_left###-->Left box content comes here<!--###boxes_left###-->...
[9] <div id="main_content"><!--###main_content###-->Main content comes here<!--###main_conte...
[10] <div id="boxes_right"><!--###boxes_right###-->Ri box content comes here<!--###boxes_rig...</div>
[11] <div id="footer"><p><br />©2011 <a href="http://www.fryske-akademy.nl/">...</div>
[12] <noscript><style type="text/css">.ifNoJS { display: none; } </style></noscript>
```

[Rows 1-4] <div> element with id “header” has three selected elements. Logo, search, and sitemap –elements are skipped in this example because it would cause repetition of the same functionality.

[2-3] As was specified, the two font-resize buttons are first introduced here. They are given IDs as well as a class ".ifNoJS" that refers to the code on row 13. The "resizeFont(x)" calls "resizeFont"-named JavaScript to be executed and feeds the specified value to it.

[4] The websites' main menu will be placed here dynamically later. "<!--###main\_menu###-->" is a reference tag for TYPO3 to recognize this element. When the dynamic content will be inserted, the "Main menu comes here" text is replaced.

[5-6] The complete "lang\_loc" (Language & Location) element includes the language switch flags and the breadcrumb trail. They all have reference tags around them because they will be dynamically filled as the "main menu" element explained above. Only one of the flags is included in this example, because the functionality is the same for each.

[7-10] The "container"-element has three elements into which the actual page content from TYPO3 will be dynamically placed.

[11] "The footer" elements are not given reference tags around them (as the two font resize elements earlier) because they are intended to be static elements meaning that we don't want TYPO3 to touch them.

[12] The only non-<div> element in the HTML is the <noscript> element which tells the website visitor's browser what to do if there is no JavaScript available. What this code basically does is that by using CSS-style definition "display: none" the elements with a class ".ifNoJS" are hidden.

The main aspect here to catch is that all <div> elements are given a specific "ID". These IDs are used as identifiers in CSS styling to control the visual appearance. The "<!--###XXX###-->" reference marker and the labels are even more important. The markers are actually the basic magic of using TYPO3: The page content of the requested page is inserted from the TYPO3's database between the tag markers specified above.

Thus now we need to look into the TypoScript code in order to fully understand the logic. The amount of TypoScript is rather long so it is divided into seven logically advancing segments. The code snippets might have parts in them that are difficult to grasp at the first glance even if there are comments, but by following the text that follows the code, it should come clear.

### *TypoScript, part 1: the basic configuration settings*

```
[1] config.doctype = xhtml_trans
[2] config.xhtmlDoctype = xhtml_trans
[3] config.xmlprologue = none
[4] page.includeCSS.file1 = style.css, page.includeJS.file1 = resizeFont.js
```

[1-2] The HTML doc type that TYPO3 should output is defined here. "xhtml\_trans" translates to "XHTML 1.0 Transitional" which is not the latest standard. But the latest standard, as of writing this report, that TYPO3 supported was XHTML 1.1, and it was attempted to get to function. But XHTML 1.1-output of TYPO3 did not properly validate so it was chosen as a better choice to keep the output XHTML 1.0, because it gets validates perfectly. Thus it was chosen as better choice to use older standard which validates rather than a more recent which does not.

[3] Using XML prologue is “a practice that is recommended but not required by the W3C”. During testing there were no issues noticed with any browsers whether this setting was on or off. Internet was searched for recommendations on this, and it was found safest solution to not include it. (Web Standards Project, 2011)

The last row shows how external CSS style sheet and JavaScript –files are instructed to be included using the TypoScript.

#### *TypoScript, part 2: the TEMPLATE-object*

This is one of the crucial parts to get right in the TypoScript in order to TYPO3 show anything at all. The HTML template file needs to be introduced to the TYPO3 in order to instruct the content as introduced in the earlier section.

```
[1] page = PAGE #===configures the HTML-template into which content will be forwarded to
[2] page.10 = TEMPLATE
[3] page.10.template = FILE
[4] page.10.template.file = template.html
```

The main thing to understand here is that “page” is just a given label, but “PAGE” tells TYPO3 to handle the “page” internally as a page (HTML output). In this case the website design is kept simple, so this is the only template defined where content will be inserted into.

Number “10” and “TEMPLATE” tell TYPO3 that at position 10 of the template page, the dynamic content will be inserted. The position number can be any number but it defines in which order TYPO3 fills up the content into the HTML template. This can make difference when the website is more complex, but with the websites in question this random number just has to be there. Nonetheless, this random number has to be used consistently in the following parts.

#### *TypoScript, part 3: image link for language switching*

This part configures what happens when an image link to switch language is pressed.

```
[1] page.10.subparts.FLAGFY = COA #===creates COA for FLAGFY
[2] page.10.subparts.FLAGFY { #===this is Frisian flag's default state when English or Dutch is active
[3] 10 = TEXT
[4] 10.data = field:uid
[5] 10.wrap = <a href="index.php?L=8&id=|
[6] 20 = IMAGE
[7] 20.altText = FY
[8] 20.titleText = Omskeakelje nei Frysk
[9] 20.file = flag-fy.png
[10] 20.wrap = ">|</a>      }
```

[1-10] These rows show how the language flags are configured to their default position when the website's default language (English) is active. Only the Frisian flag is shown because the script itself is the same for each.

[1-2] The marker FLAGFY, which was introduced in the HTML template, is now used here. The row first row creates an internal TYPO3 cObject COA (content object array) named “FLAGFY” which is the same as the label of the marker. “page.10.subparts.FLAGFY” basically tells TYPO3 to find “FLAGFY”-marker in the HTML-template file and then the stuff that is defined on rows 3-10 is inserted there.

[3-5] The position number in the beginning of the row defines the order of execution. Row 4 tells the TYPO3 to use the "uid" (currently open page's unique TYPO3-ID) and on row 5 the "uid"-variable is wrapped into the "|" position of the text. So the text outputted by row 5 is "<a href='index.php?L=8&id=uid'".

[6-10] A similar action to rows 3-5 is defined here, but image element <img> is wrapped instead. The image is given important accessibility attributes *alt* and *title* on rows 7-8, and the whole element is then wrapped to position "|" on row 10.

The end result output from rows 3 and 5 combined is <a href="index.php?L=8&id=uid"></a>. This is dynamically created image link where the *uid* is the currently open page's unique ID number. The "L=8"-variable in the static code refers to the system language ID number, which in this specific case is defined as number 8 for Frisian language.

The basis script for the other languages is the same, only the values are different.

#### *TypoScript, part 4: dynamic language switch –link*

When the default language is switched on multilingual TYPO3 website by the site visitor, the following things must be configured in the site's template to make translated content to appear correctly site wide:

```
[1] [globalVar=GP:L=8] #==the settings below activate when the requested URL has L=8
[2] config.sys_language_uid=8
[3] config.language=fy
[4] config.locale_all = nl_NL
[5] page.10.subparts.FLAGFY.20.file = flag-fy_act.png
```

[1] The first row simply tells that the below settings are applied only when the website visitor's URL includes language-tag with certain value, in this case "L=8".

[2-4] These rows instruct TYPO3 to switch the language. This makes the translated versions of the pages to be displayed instead of the default. Because TYPO3 has no support for Frisian internally, it has to be configured manually into this TYPO3 installation separately. However, the setting defined on the row 4 can't be customized, that is why the closest match, Dutch, is used.

[5] This setting replaces the default image file set on the previous page (row 9) to a image that shows visually that Frisian language is now active.

#### *TypoScript, part 5: main menu HTML generation*

TYPO3 comes with internal support to (automatically) output the titles of the page names for creating navigational menu, and this implementation is also well documented (Skårhøj, 2002), thus only the highlighted part of the code (that is relevant to this report) is explained below.

```
page.10.subparts.main_menu = HMENU
page.10.subparts.main_menu.1 = TMENU
page.10.subparts.main_menu.1 {
wrap = <ul class="mainmenu">|</ul> #===class for the top level is defined
NO.ATagTitle.field = title
NO.stdWrap.htmlSpecialChars = 1
NO.wrapItemAndSub = <li class="menu_11item">|</li> #===menu items get a class label...
ACT = 1
ACT.ATagTitle.field = title
```

```

ACT.stdWrap.htmlspecialchars = 1
ACT.wrapItemAndSub = <li id="active-I1">|</li> } #===but active menu item gets specific id
page.10.subparts.main_menu.2 = TMENU
page.10.subparts.main_menu.2 {
wrap = <ul class="menu_level1">|</ul> #===class for the 1st sublevel is defined
NO.ATagTitle.field = title
NO.stdWrap.htmlspecialchars = 1
NO.wrapItemAndSub = <li class="menu_I2item">|</li> } #===all menu items get a class label...
ACT = 1
ACT.ATagTitle.field = title
ACT.stdWrap.htmlspecialchars = 1
ACT.wrapItemAndSub = <li id="active-I2">|</li> } #===but active menu item gets specific id

```

In order to be able to apply CSS styling to the menu (was necessary in order to replicate the old menu), the menu items need to be given certain classes and/or IDs. If the active menu item is wanted to be distinguishable (as was the case here), the above settings are necessary. Rows starting with *ACT* in front refer to menu item that is along the currently open page's path, whereas *NO* refers to non-active items.

#### *TypoScript, part 6: left and right column content*

The left and right columns on the new design layout are shown site wide, meaning that they are always visible. Moreover, it is also made possible to add left and right column content on any page which is then displayed on top of the site wide columns.

```

lib.boxesContent_left = COA
lib.boxesContent_left {
10 = CONTENT
10 {
table = tt_content #====this is the table label in the database where the content items are stored
select { #=====what content items are collected (below)
languageField = sys_language_uid #====select only in the currently active language
where = colPos=1 #====collects content only from Left (position 1 in database) column ====
orderBy = sorting #====items are shown in order
} renderObj.stdWrap.wrap = <div class="box_footer">|</div> #==every item is given additional div class
} 11 = CONTENT
11 {
table = tt_content
select {
pidInList = 99 #==== this time content items are collected from specific location instead (this way
certain content can be shown site wide)
languageField = sys_language_uid
where = colPos=1
orderBy = rand() #=====items are put in random order
} renderObj.stdWrap.wrap = <div class="box_footer">|</div>
} }

```

In order to wrap every content item into a separate box, the content items are gathered one by one, and then wrapped into additional `<div>` element class named "box footer". The name *footer* is used because in the CSS configuration this element is styled to create the rounded corner image to the final output.

At this stage, the box container was also expanded (by basically creating a new background image which is part of the CSS style sheets) so that it could contain more items than what the original Plone layout design allowed.

### *TypoScript, part 7: body tag and final content insertion*

In order to control the font resizing, a tag is inserted on every page that calls JavaScript "setFont" to be executed.

```
[1] page.bodyTag >      #====bodyTag for every page (needed for the font resize cookie)====  
[2] page.bodyTagCObject.wrap = <body onload="setFont();"> #==this executes JavaScript setFont  
[3] page.10.subparts.bboxes_left < lib.bboxesContent_left #==this inserts the content of boxes on left  
[4] page.10.subparts.main_content < styles.content.get #==this inserts the main content to template  
[5] page.10.subparts.bboxes_right < lib.bboxesContent_right #==this inserts the content of boxes on right
```

[1-2] This checks if the website visitor already has a cookie which has one font size saved out of three possible defined. If no cookie is found, the default size is set and saved.

[3-5] Here the main contents are instructed to be inserted to the HTML template that were specified in the beginning of this section (page 19).

## 4.2 CSS style sheets

The menu graphics, border, link and link hovering colors, are unique to each site, defined separately for each site apart from the master settings that all sites share. The navigation menu's visual appearance is created entirely with CSS by modifying and enhancing the original configuration from Plone template to work under TYPO3. The menu's container for the 2nd level of menu items was extended as requested by drawing a new picture and changing the values in the CSS settings accordingly.

One worth mentioning CSS code that was included during implementation is "*html {overflow-y: scroll;}*" which makes the vertical scrollbar always present on the right side. In Microsoft Internet Explorer this is a standard feature but in other browsers (this was noticed by the Fryske Akademy testers) websites' horizontal position shifts awkwardly at higher resolutions when the scrollbar isn't displayed.

The actual final CSS files are filled with relevant comments that give the sufficient information, if and when the templates need to be altered in the future.

## 4.3 Page structure and content migration

After having a template successfully configured, the content migration is resumed. The page structure of the websites was recreated inside TYPO3 first because this allowed setting the internal links between the pages even though the content was not yet migrated.

The original Plone websites' database content was accessed using the Zope interface (page 12) by opening each content element one by one. This way, even the content that was not publically available got migrated certainly from the database. It was also easy to stay organized because the content was processed in order. During process there were existing Frisian and Dutch translations of some pages found of which the current web editors had not even heard of.



Images, PDF-documents and other type of downloadable files were organized clearly into a new folder organization when uploaded to the new server.

One “fancy” content presentation-related extension was installed here. The new design of the front page of Mercator Research Centre is created with “Multiple Content” extension which allows multiple page content items to be shown at once in various ways. The one applied functions so that when one page content is displayed, only the titles of the remaining pages are shown. But these titles are actually links which, when clicked, show the actual content under the clicked title, while hiding the previously open page content. Figure 9 on page 19 shows this extension in action.

#### 4.4 Overlapping content

During implementation, there were content categories found that are shown on two different sites. The content in ‘Mercator Newsletter’, ‘Mercator News Archive’, and ‘Expert of the Month’ shown on both Mercator Research and Mercator Network websites was found to be exactly the same. In Plone this was realized by copying and pasting the same content separately into both sites.

However, TYPO3 CMS has more beneficial functions, mounting for example, to display content at other pages within the same installation. This way, the web editor can be more productive because they need to update the content only on one page, and it is then automatically updated wherever else that content is mounted to be shown. This way the possible human error of forgetting does not occur either.

During the finalizing phase, the contact info pages were also updated so that there was a completely new visitor information page created instead of showing all contact information on one page. Because the visitor information is exactly the same on each site, the actual content is stored only on the main site (Mercator Research Centre) and mounted to the two others.

#### 4.5 Accessibility improvements

The font resizing accessibility feature was implemented. Instead of defining three specific sizes as on Fryske Akademy website, the design was changed to two button logic where the font size can simply only be increased or decreased, but not without limitations though. The code for the font resize is found in the appendix 5.

The font resizing is the only feature that has been implemented in JavaScript. As was explained in the implementation chapter; in case the site visitor has no JavaScript at all, or it is disabled, the visitor does not even see the buttons. This avoids the negative browsing experience resulting from a non-functioning feature.

Other minor but effective visual improvements were implemented as they were requested. Most noticeably, the font size of the main menu items was increased.

## 4.6 Search Engine Optimization

The preservation of the speaking URLs turned out to be a technical challenge in the end, but it was a beneficial TYPO3 and network configuration learning experience for the IT department of Fryske Akademy. The RealURL installation and configuration was a challenge due to complex network configurations. During the installation process, a lot of redundant hosting and re-direction configuration got cleaned up, and this was a prerequisite for RealURL to function flawlessly.

Another installed SEO related extension was the "Basic SEO Features". It "introduces a separate field for the title tag per page, easy and SEO-friendly keywords and description editing in a new module as well as a Google Sitemap (XML) and a clean output in the HTML source code" (Mack, 2011).

## 5 Evaluation

### 5.1 Markup validation

The goal was set that websites' HTML and CSS outputs should pass the W3C's Markup Validation Service test. The final actual HTML output gets otherwise full validity but there is one error that is due to the trick that hides the font resize elements if there is no JavaScript (row 12 of HTML-template, page 19). Nonetheless, this "error" does not make the websites less accessible at all.

CSS validation is 100 % on CSS3-level. CSS2.1-level validation results in one error because the earlier mentioned CSS attribute for forcing the vertical scrollbar to be displayed is a CSS3 attribute.

### 5.2 Browser compatibility

There are numerous (free) applications and techniques for web browsing available, and testing a website with all of them is a daunting task, but I took it as a habit during the project to perform testing of the new websites using different browsers at different locations with different operating systems and hardware setups even.

Thanks to e.g. IETester application, it is fast to test websites' functionality in older Microsoft Internet Explorer (IE) versions locally. It was found that the visibility of the main menu's sublevel does not work in versions 5.5 and 6 of IE due to known programming errors. But instead of applying some questionable CSS hacks to make it work as before, the websites' content was rearranged so that it is not necessary to even see the sublevels of the main menu items, because they are now presented in the main content container of the website when the main menu item is clicked.

Test browsing with mobile devices (such as iPhone and some Nokia Symbian based cell phones) was performed with actual devices and using simulators and emulators available online such as [iphonetester.com](http://iphonetester.com).

There were no incompatible browsers found except the mentioned old versions of IE. The text based browsers (no CSS styling) were also tested and found to present the websites' content in logical and understandable order.

### 5.3 Content integrity

Due to the method used for content copying, the contents' integrity was instantly verified. But to make sure, it was done a second time manually by having both the old and new website open simultaneously in two separate browser windows. Then every page of every site was browsed and the integrity was then witnessed.

There are lots of (free) applications available to verify websites' functionality. For example "Xenu's Link Sleuth" was found to be very useful tool for verifying functionality of the links and images.

Non-public content pages (the "lost" Dutch and Frisian translations of old pages) and some random files were also copied from the old database to the new file directory under TYPO3.

## 6 Conclusions and further study ideas

To have an appropriate approach to designing the website migration implementation, it is a prerequisite to know whether it is the intention to simply clone the sites to another platform, or if improvements to the design are expected as well. The content transfer is a mechanical task for the most part, whereas re-creating the template design involves many decisions to be made.

A “keep it technically as simple as possible” mentality in planning the new design will usually result in less time consuming maintenance and allows more flexible improvement possibilities in the end, when it comes to adjusting the design according to client’s requirements or wishes. This resulted to be true in this project.

Despite the fact that 1) all project objectives in this case were achieved, although 2) visual appearance of the websites was improved, and even though 3) some accessibility features and 4) some search engine optimizations were implemented, room for improvement always remains. The visual design choices are often the hardest as it is hard to please everyone. One can dive into world of never-ending choices with each component of website design possibilities.

Spending time to do extensive search engine optimizations can be considered even a bit questionable manipulation of the “system”. The benefits are hard to measure. Implementing every possible accessibility feature defined in the guidelines is of course possible but again, how to quantify would it be worth the effort?

So perhaps the website design development field should push into direction of standardization that would make external accessibility enhancements more compatible with the website designs, instead of every web developer making their own version of designing and implementing each accessibility feature?

### 6.1 Personal reflections

Maintaining a website, especially from a technical design perspective, can be classified as a never-ending task because (technical) standards keep changing and designs can be implemented in countless ways. Once get the template design adjusted well according to client’s wishes, it is better not to alter it unless the need for change is analyzed and tested well enough. It requires careful studying and verifying to find out which settings are safest to implement without breaking something in the website overall.

Maybe it would be fine to not care too much about guidelines and consistency, and just put in all sorts of hacks in and fix problems as they come along. I can’t help belonging to the standard following crowd though. Once get it right, one can peacefully leave it be and even if something needs to be adjusted sometimes, because it was designed to be clear as possible to understand, it makes the webmaster’s life less stressful. Until the next need for change arises and the problem

of migrating the content to be accessible under different platform is faced again. However, a wise (and sometimes lucky) choice with the CMS software can avoid the need to do platform-migrations such as this entirely.

As a more managerial than programming oriented engineering student I couldn't help approaching this assignment more from my own educational perspective, especially by taking into account the time constraint. Programming a script for content copying purpose seemed unbeneficial task next to realizing how to design a functional standard-compliant template for TYPO3. Sometimes, rather than spending a great deal of time to invent a script that would maybe do most of the job, but in a project in a scale of this one, the manual method is more sensible because it allows being more in control and in the end; to verify personally that all content is present as before and displayed properly under the new majorly altered design.

The database structures could certainly have been explored further, so that the content would have been copied in a more engineer-type method, but mainly a lack of time was the biggest enemy during this project. Attempts were made to extract the data objects from ZODB to a format that could have been transformed into TYPO3's database. Even if I would have been able to copy all the data with a script-method, I know I would have gone through every single page manually anyway to verify their integrity under the new platform, and moreover, to fix the bad HTML code to suit the new design requisites, and to update the page content items (images, downloadable files) to their new locations.

Even my attempts to invent a script-method for this migration were unsuccessful it certainly can be accomplished. But is it worth the effort, it needs to be quantified against the other odds and challenges at hand, because every migration project is unique.

## References

Altmann, W., Fritz R. and Hinderink D., 2005. *TYPO3: Enterprise Content Management*. Birmingham, United Kingdom: Packt Publishing Ltd.

Anglia Ruskin University Library, 2011. *Harvard System of Referencing Guide*. [online] Available at: <<http://libweb.anglia.ac.uk/referencing/harvard.htm>> [Accessed 15 August 2011].

Battles, Ryan, 2008. *Is a Content Management System Right for You?* [blog] December 23, Available at: <[http://joviawebstudio.com/index.php/blog/is\\_a\\_content\\_management\\_system\\_right\\_for\\_you/](http://joviawebstudio.com/index.php/blog/is_a_content_management_system_right_for_you/)> [Accessed 15 August 2011].

Coles, Malcolm, 2009. *SEO friendly URLs: myth and fact*. [blog] February 7, Available at: <<http://www.malcolmcoles.co.uk/blog/seo-friendly-urls-myth-and-fact/>> [Accessed 15 August 2011].

Fryske Akademy, 2011. [online] Available at: <<http://www.fryske-akademy.nl/>> [Accessed 15 August 2011].

Furrer, J., 2010. *Manual for Multiple Content*. [online] Available at: <<http://typo3.org/documentation/document-library/extension-manuals/jfmulticontent/1.5.1/view/toc/0/>> [Accessed 15 August 2011].

Gift, N. & Rhodes, B.C., 2008. *Example-driven ZODB*. [online] Available at: <<http://www.ibm.com/developerworks/aix/library/au-zodb/>> [Accessed 15 August 2011].

Gomes, D. and Silva, M.J., 2006. *Modelling Information Persistence on the Web*. [online] Available at: <<http://xldb.di.fc.ul.pt/daniel/docs/papers/gomes06urlPersistence.pdf>> [Accessed 15 August 2011] (Archived by WebCite® at <http://www.webcitation.org/612dc8508>).

Mack, B., 2011. *Basic SEO Features manual*. [online] Available at: <[http://typo3.org/documentation/document-library/extension-manuals/seo\\_basics/0.6.0/view/toc/0/](http://typo3.org/documentation/document-library/extension-manuals/seo_basics/0.6.0/view/toc/0/)> [Accessed 15 August 2011].

Mercator Network, 2011. [online] Available at: <<http://www.mercator-network.eu/>> [Accessed 15 August 2011].

Mercator Network of Schools, 2011. [online] Available at: <<http://www.networkofschools.eu/>> [Accessed 15 August 2011].

Mercator Research Centre, 2011. [online] Available at: <<http://www.mercator-research.eu/>> [Accessed 15 August 2011].

Microsoft, 2011. *The Internet Explorer 6 Countdown*. [online] Available at: <<http://www.ie6countdown.com/>> [Accessed 15 August 2011].

Plone, 2011. *ZODB Database*. [online] Available at: <<http://plone.org/documentation/manual/plone-community-developer-documentation/persistence/database>> [Accessed 15 August 2011].

Quintagroup, 2010. *Content migration from Plone 2.1 to Plone 3.2*. [online] Available at: <<http://projects.quintagroup.com/products/wiki/quintagroup.transmogrifier/plone2-3>> [Accessed 15 August 2011].

Skårhøj, Kasper, 2004. *Futuristic Template Building*. [online] Available at: <[http://typo3.org/documentation/document-library/tutorials/doc\\_tut\\_ftb1/current/](http://typo3.org/documentation/document-library/tutorials/doc_tut_ftb1/current/)> [Accessed 15 August 2011].

Skårhøj, Kasper, 2003a. *Modern Template Building, Part 1*. [online] Available at: <[http://typo3.org/documentation/document-library/tutorials/doc\\_tut\\_tmplselect/current/](http://typo3.org/documentation/document-library/tutorials/doc_tut_tmplselect/current/)> [Accessed 15 August 2011].

Skårhøj, Kasper, 2003b. *Modern Template Building, Parts 2-3*. [online] Available at: <[http://typo3.org/documentation/document-library/tutorials/doc\\_tut\\_tmplselect2/current/](http://typo3.org/documentation/document-library/tutorials/doc_tut_tmplselect2/current/)> [Accessed 15 August 2011].

Skårhøj, Kasper, 2002. *TypoScript by example*. [online] Available at: <[http://typo3.org/documentation/document-library/tutorials/doc\\_core\\_tsbeyex/current/](http://typo3.org/documentation/document-library/tutorials/doc_core_tsbeyex/current/)> [Accessed 15 August 2011].

Taylor, M.J., 2007. *Perfect multi-column CSS liquid layouts*. [blog] October 27, Available at: <<http://matthewjamestaylor.com/blog/perfect-multi-column-liquid-layouts>> [Accessed 15 August 2011].

TYPO3, 2011. *The Enterprise Open Source CMS*. [online] Available at: <<http://typo3.org>> [Accessed 15 August 2011].

W3C, 2011. *Web Content Accessibility Guidelines (WCAG)*. [online] Available at: <<http://www.w3.org/WAI/intro/wcag>> [Accessed 15 August 2011].

Web Standards Project, 2011. *Browser Problems with the XML Prolog*. [online] Available at: <[http://www.webstandards.org/learn/articles/prolog\\_problems/](http://www.webstandards.org/learn/articles/prolog_problems/)> [Accessed 15 August 2011].

## APPENDICES to “Plone® to TYPO3®”

|   |                                                                            |    |
|---|----------------------------------------------------------------------------|----|
| 1 | File directory structure .....                                             | 2  |
| 2 | HTML template.....                                                         | 3  |
| 3 | TypoScript.....                                                            | 4  |
| 4 | CSS style sheet.....                                                       | 8  |
| 5 | JavaScripts .....                                                          | 14 |
| 6 | Manuals .....                                                              | 15 |
|   | How to add content to “boxes” and the limitations.....                     | 15 |
|   | Network of Schools site’s “clickable map” -manual .....                    | 16 |
|   | How to add/remove/modify Calendar events .....                             | 17 |
|   | How to add link to upcoming event to the box .....                         | 17 |
|   | How to use Multiple Content –plugin extension .....                        | 17 |
|   | Visitor information –page needs to be updated only on one page.....        | 18 |
|   | Good to know about deleted page content items.....                         | 18 |
|   | In case the main menu items are modified.....                              | 19 |
|   | Main numerical design-limitations collected in one from all the above..... | 19 |



# 1 File directory structure

This page describes the file directory structure and where the template design files (HTML, TypoScript, CSS and JavaScript –files) are stored.

- Every site has its own folder (“mercator”, “network”, and “nos”) under the “fileadmin/”-directory, that is accessible via TYPO3 –backend. This is where all external files are stored.
- Every site’s TypoScript-code is individual and named as “typoscr.txt”.
  - Same goes for the HTML-template (template.html).
- “mercator”-directory holds the master CSS-settings file (style.css) and JavaScript-files (ReadCookie.js, resizeFont.js, SetCookie.js and setFont.js) - these are used by all sites.
- Every site’s individual CSS style sheet’s filename is clear:
  - “style-mercator.css” “style-network.css” and “style-nos.css”.
- All image-files are saved under “images” directory under each site’s own directory.
- Names for the directories correspond to the actual page-name titles.

## 2 HTML template

File name: [template.html] (each site has its own file.)

```
<div id="header">
  <div id="merclogo"><a href="/"></a></div>
  <div id="search">
    <form action="index.php?id=1551&no_cache=1" id="mailform"
    enctype="multipart/form-data" method="post" onsubmit="return
    validateForm('mailform','','','')">
      <div style="display:none;">
        <input type="hidden" name="stype" id="mailformstype" value="L0" />
        <input type="hidden" name="locationData" value="359:tt_content:813" />
      </div>
      <div class="csc-mailform-field">
        <label for="mailformsword"></label>
        <input type="text" name="sword" id="mailformsword" size="10" value="" />
        <input type="hidden" name="scols" id="mailformscols" value="pages.title-subtitle-
keywords-description:tt_content.header" />
      </div>
    </form>
  </div>
  <div id="part-of-fa"><a href="http://www.fryske-akademy.nl/" target="_new"></a></div>
  <div id="fontbigger" class="ifNoJS"><a href="#" onclick="resizeFont(1);"></a></div>
  <div id="fontsmaller" class="ifNoJS"><a href="#" onclick="resizeFont(-1);"></a></div>
  <div id="sitemap"><a href="/home/sitemap/"></a></div>
  <div id="headerCornerUp"></div>
  <div id="headerCorner"></div>
  <div id="main_menu"><!--###main_menu###-->Main menu comes here<!--###main_menu###--
></div>
</div>

<div id="lang_loc">
  <div id="flagfy"><!--###FLAGFY###-->Frysk flag comes here<!--###FLAGFY###--></div>
  <div id="flagnl"><!--###FLAGNL###-->Dutch flag comes here<!--###FLAGNL###--></div>
  <div id="flagen"><!--###FLAGEN###-->English flag comes here<!--###FLAGEN###--></div>
  <div id="location"><!--###LOCATION_CURRENT###-->U are here:<!--
###LOCATION_CURRENT###--></div>
</div>
```

Document version:  
2011 August 22

```

<div id="container">
  <div id="boxes_left"><!--###boxes_left###-->Left box content comes here<!--
###boxes_left###--></div>
  <div id="main_content"><!--###main_content###-->Main content comes here<!--
###main_content###--></div>
  <div id="boxes_right"><!--###boxes_right###-->Right box content comes here<!--
###boxes_right###--></div>
</div>

<div id="footer"><p><br />©2011 <a href="http://www.fryske-akademy.nl/en/"
target="_blank" title="Fryske Akademy">Fryske Akademy</a></p></div>

<noscript>
  <style type="text/css">
    .ifNoJS { display: none; }
  </style>
</noscript>

```

### 3 TypeScript

File name: [typoscr.txt] (each site has its own file.)

```

#====Default configuration====
config.doctype = xhtml_trans
config.xhtmlDoctype = xhtml_trans
config.xmlprologue = none
config.xhtml_cleaning = all
config.language=en
config.locale_all = en_GB
page.shortcutIcon = fileadmin/mercator/images/favicon.ico // SITE-SPECIFIC
#====CSS files====
page.includeCSS.file1 = fileadmin/mercator/style.css
page.includeCSS.file2 = fileadmin/mercator/style-mercator.css // SITE-SPECIFIC CSS
page.includeCSS.file3 = fileadmin/mercator/calendar.css // ONLY MERCATOR-RESEARCH
HAS CALENDAR, THESE SETTINGS MAKE IT LOOK LIKE THE CALENDAR ON FRYSKE AKADEMY-SITE
#====JavaScripts for Font resize and cookie====
page.includeJS.file1 = fileadmin/mercator/resizeFont.js
page.includeJS.file2 = fileadmin/mercator/setFont.js
page.includeJS.file3 = fileadmin/mercator/SetCookie.js
page.includeJS.file4 = fileadmin/mercator/ReadCookie.js
#====RealURL stuff====
config.baseURL = http://www.mercator-research.eu/ // SITE-SPECIFIC SETTING
config.tx_realurl_enable = 1
config.simulateStaticDocuments = 0
config.prefixLocalAnchors = all
config.uniqueLinkVars = 1
config.linkVars = L

```

Document version:  
2011 August 22

```
config.defaultToHTMLSuffixOnPrev = 1

#====global keywords===== // UNIQUE TO EACH SITE
page.meta.keywords = mercator, mercator research, fryske akademy, frisian academy,
frisian, fryske, frysk, mercator european research centre, multilingualism, language
learning, mercator research centre
page.meta.keywords.override.field = keywords

#====Language and Location control===== // ONLY MERCATOR-RESEARCH HAS ADD. LANGUAGES
#=set flags to default=
page.10.subparts.FLAGFY = COA
page.10.subparts.FLAGFY {
10 = TEXT
10.data = field:uid
10.wrap = <a href="index.php?L=8&id=|
20 = IMAGE
20.altText = FY
20.titleText = Omskeakelje nei Frysk
20.file = fileadmin/mercator/images/template/flag-fy.png
20.wrap = ">|</a>
}
page.10.subparts.FLAGNL = COA
page.10.subparts.FLAGNL {
10 = TEXT
10.data = field:uid
10.wrap = <a href="index.php?L=4&id=|
20 = IMAGE
20.altText = NL
20.titleText = Verander taal in Nederlands
20.file = fileadmin/mercator/images/template/flag-nl.png
20.wrap = ">|</a>
}
page.10.subparts.FLAGEN = COA
page.10.subparts.FLAGEN {
10 = TEXT
10.data = field:uid
10.wrap = <a href="index.php?L=5&id=|
20 = IMAGE
20.altText = EN
20.titleText = Switch language to English
20.file = fileadmin/mercator/images/template/flag-gb_act.png
20.wrap = ">|</a> }

#====breadcrumbs (default English)=====
page.10.subparts.LOCATION_CURRENT=HMENU
page.10.subparts.LOCATION_CURRENT {
special=rootline
special.range= 1 | -1
wrap = You are here:&nbsp;  |
1 = TMENU
1.NO.allWrap = |      &nbsp;  &rarr;&nbsp;  | *|      &nbsp;  &rarr;&nbsp;  | *|      | }
```

**####adjusts the presentation and breadcrumbs to current language selection==== // ONLY  
MERCATOR-RESEARCH HAS ADDITIONAL LANGUAGES**

```
[globalVar=GP:L=8]
config.sys_language_uid=8
config.language=fy
config.locale_all = nl_NL
page.10.subparts.LANGUAGE_CURRENT = TEXT
page.10.subparts.LANGUAGE_CURRENT.value = Frysk
page.10.subparts.LOCATION_CURRENT.wrap = Jo binne hjir:&nbsp;|
page.10.subparts.FLAGFY.20.file = fileadmin/mercator/images/template/flag-fy_act.png
page.10.subparts.FLAGNL.20.file = fileadmin/mercator/images/template/flag-nl.png
page.10.subparts.FLAGEN.20.file = fileadmin/mercator/images/template/flag-gb.png
[end]
```

```
[globalVar=GP:L=4]
config.sys_language_uid=4
config.language=nl
config.locale_all = nl_NL
page.10.subparts.LANGUAGE_CURRENT = TEXT
page.10.subparts.LANGUAGE_CURRENT.value = Nederlands
page.10.subparts.LOCATION_CURRENT.wrap = U bent hier:&nbsp;|
page.10.subparts.FLAGFY.20.file = fileadmin/mercator/images/template/flag-fy.png
page.10.subparts.FLAGNL.20.file = fileadmin/mercator/images/template/flag-nl_act.png
page.10.subparts.FLAGEN.20.file = fileadmin/mercator/images/template/flag-gb.png
[end]
```

```
[globalVar=GP:L=5]
config.sys_language_uid=5
config.language=en
config.locale_all = en_GB
page.10.subparts.LANGUAGE_CURRENT = TEXT
page.10.subparts.LANGUAGE_CURRENT.value = English
page.10.subparts.LOCATION_CURRENT.wrap = You are here:&nbsp;|
page.10.subparts.FLAGFY.20.file = fileadmin/mercator/images/template/flag-fy.png
page.10.subparts.FLAGNL.20.file = fileadmin/mercator/images/template/flag-nl.png
page.10.subparts.FLAGEN.20.file = fileadmin/mercator/images/template/flag-gb_act.png
[end]
```

**####TypoScript for Horizontal 2-Level Menu (main\_menu) cObject====**

```
page.10.subparts.main_menu = HMENU
page.10.subparts.main_menu.1 = TMENU
page.10.subparts.main_menu.1 {
expAll = 1
noBlur=1
wrap = <ul class="menu_2">|</ul>
NO.ATagTitle.field = title
NO.stdWrap.htmlSpecialChars = 1
NO.wrapItemAndSub = <li class="menu2_llitem">|</li>
ACT = 1
ACT.stdWrap.htmlSpecialChars = 1
ACT.wrapItemAndSub = <li id="active-11">|</li>
ACT.ATagTitle.field = title }
```

```

page.10.subparts.main_menu.2 = TMENU
page.10.subparts.main_menu.2 {
expAll = 1
NO.ATagTitle.field = title
NO.stdWrap.htmlSpecialChars = 1
wrap = <ul class="menu2_level1"> | </ul>
NO.wrapItemAndSub = <li class="menu2_l2item"> | </li>
ACT = 1
ACT.stdWrap.htmlSpecialChars = 1
ACT.wrapItemAndSub = <li id="active-l2"> | </li>
ACT.ATagTitle.field = title
}
#####This Typoscript is for the Boxes on the left side=====
lib_boxesContent_left = COA
lib_boxesContent_left {
10 = CONTENT
10 {
table = tt_content
select {
languageField = sys_language_uid
where = colPos=1 #####collects all content from Left column on Current page=====
orderBy = sorting #####can switch to 'rand()' if want random order=====
}
renderObj.stdWrap.wrap = <div class="box_footer">|</div>
}
11 = CONTENT
11 {
table = tt_content
select {
pidInList = 1804 #####UID of the "#Boxes"-folder where the content elements are
stored that are shown on every page of the site=====
languageField = sys_language_uid
where = colPos=1
orderBy = sorting
}
renderObj.stdWrap.wrap = <div class="box_footer">|</div>
}
}
#####This Typoscript is for Boxes on right side, identical to above except for colPos
lib_boxesContent_right = COA
lib_boxesContent_right {
10 = CONTENT
10 {
table = tt_content
select {
languageField = sys_language_uid
where = colPos=2
orderBy = sorting }
renderObj.stdWrap.wrap = <div class="box_footer">|</div> }
11 = CONTENT
11 {
table = tt_content

```

```

select {
    pidInList = 1804
    languageField = sys_language_uid
    where = colPos=2
    orderBy = rand()
}
renderObj.stdWrap.wrap = <div class="box_footer">|</div>
}
}
#====Content -elements are configured here====
page.10.subparts.bboxes_left < lib.bboxesContent_left
page.10.subparts.main_content < styles.content.get
page.10.subparts.bboxes_right < lib.bboxesContent_right

#====bodyTag for every page (needed for the font resize cookie)====
page.bodyTag >
page.bodyTagCObject = TEXT
page.bodyTagCObject.data = field:id
page.bodyTagCObject.wrap = <body class="fontti" onload="setFont();">

#====Default PAGE object (html-template into which all above stuff is inserted into)==
page = PAGE
page.10 = TEMPLATE
page.10 {
    template = FILE
        template.file = fileadmin/mercator/template.html
}

```

## 4 CSS style sheet

There is one crucial CSS-setting for main navigational menu in case the changes are made to the names of the main page-titles: `#main_menu {margin-left: XXXpx;} /* MENU'S DISTANCE FROM LEFT!!`

The setting is clearly marked (as above) and easy to find in a separate CSS-style sheet for each site.

`style.css` // Master CSS for all sites (located in "mercator" directory)

```

html {overflow-y: scroll;} /* scroll-bar is always present on the right */

/* Links and hr (colors are set separately in each site's own CSS) */
a:link { text-decoration: none; background-color: transparent; }
a:hover{ text-decoration: underline; background-color: transparent; }
a:visited{ text-decoration: none; background-color: transparent; }

```

```

hr { height: 1px; margin: 0.5em 0 1em 0; }

/* This is to ensure that when you align texts in Typo3 RTE they do get aligned */
.align-left {text-align: left; }
.align-center {text-align: center; }
.align-right {text-align: right; }

/* Header (includes Search & Fontsize-select) */
#header { width: 1000px; height: 147px; position: relative; padding-left:
0px;overflow: visible; background:White; margin:0px auto; text-align: left; }
#search { position: absolute; top: 40px; right: 20px; }
#search DIV.csc-mailform-field { font-size: x-small; }
#fontbigger { position: absolute; top: 105px; right: 42px; font-size: 140% !important;
}
#fontbigger a:link { color: #000000 !important; }
#fontbigger a:hover{ color: #000000 !important; text-decoration: underline; }
#fontbigger a:visited{ color: #000000 !important; }
#fontsmaller { position: absolute; top: 95px; right: 30px; font-size: 100% !important;
}
#fontsmaller a:link { color: #000000 !important; }
#fontsmaller a:hover{ color: #000000 !important; text-decoration: underline;}
#fontsmaller a:visited{ color: #000000 !important; }
#sitemap { position: absolute; right: 20px; top: 10px; }
#sitemap a:link { color: #000000 !important; }
#sitemap a:hover{ color: #000000 !important; text-decoration: underline; font-size:
100% !important;}
#sitemap a:visited{ color: #000000 !important; }
#language { position: absolute; padding-left: 50px; top: 140px; font-family: Verdana;
font-size: 11px; }
#headerCornerUp { position: absolute; right: 0px; top: 0px; }
#headerCorner { position: absolute; right: 0px; top: 140px; }

/* Language&Location */
#lang_loc { width: 1000px; padding-top: 1px; margin-top: 15px; margin-bottom: 0px;
margin-left: auto; margin-right: auto; text-align: left; font-size: 11px; font-family:
Verdana; overflow: visible; background: White; height: 16px; border-top: 0px dashed;
border-bottom: 0px dashed; }
/* Language selector flags size & positioning */
#flagfy IMG, #flagnl IMG, #flagen IMG { float: left; width: 20px; height: 15px;
margin-left: 0px; margin-right: 5px; }
#flagfy IMG { padding-left: 50px; }
/* Location(you are here) */
#location { float: left; }

/* Main container DIV ("content"+"boxes" are inside of this DIV) */
#container { width: 1000px; margin-left: auto; margin-right: auto; margin-top: 50px;
text-align: left; overflow: visible; }

/* Main content DIV */
#main_content { width: 640px; overflow: hidden; padding-left: 25px; padding-right:
25px;
vertical-align: top; float: left; background: White; height: auto !important; min-
height: 300px; }

```



```

#main_content .csc-mailform { border-style: groove; border-width: 2px; }
#main_content h1, h2, h3, h4, h5, h6, p, ul { padding-left: 15px; }

/* "Boxes" */
#boxes_left { float: left; vertical-align: top; width: 154px; margin-left: 2px; }
#boxes_right { float: right; vertical-align: top; width: 154px; margin-right: 0px; }
#boxes_left DIV.csc-header, #boxes_right DIV.csc-header { margin-top: 10px; padding: 0
!important; border-top: none; border-left: none; border-right: none;
    background: url("images/template/portlet_header.gif") no-repeat top; text-align:
center; }
#boxes_left DIV.csc-header H1, #boxes_right DIV.csc-header H1
    { font-size: x-small; border-bottom: none; padding-top: 2px; padding-left: 0; font-
weight: normal; margin: 0; /*text-decoration: blink;*/ }
#boxes_left DIV.csc-default p, #boxes_right DIV.csc-default p
    { font-size: x-small; border-bottom: none; margin-left: 5px; margin-right: 5px;
padding-top: 2px; padding-left: 0; /*text-decoration: blink;*/ }
#boxes_left DIV.csc-default ul, #boxes_right DIV.csc-default ul
    { font-size: x-small; border-bottom: none; padding-left: 15px; margin-right: 5px;
padding-top: 2px; /*text-decoration: blink;*/ }
#boxes_left img, #boxes_right img { margin-top: 4px; margin-left: 2px; margin-right:
2px; }
.box_footer { background: url("images/template/portlet_footer_inv_new.png") no-repeat
bottom; padding-bottom: 5px; border: 0;}

/* Footer */
#footer P { font-size: x-small; text-align: center; vertical-align: bottom; }
/*#footer { border-top: 1px dashed; clear: both; }*/

/* All Images - No Borders (IE) + no margin */
img { border: 0; margin: 0; }

/* Horizontal Menu */
#main_menu {margin-top: 120px; min-height: 21px; max-height: 21px; border:
0; background: transparent; list-style: none;
    position: absolute; min-width: 800px; max-width: 800px;
overflow: visible;}
.menu_2 {border: 0; background: transparent; list-style: none; height:
21px; font-family: Verdana; font-size: 12px; }
.menu_2 ul {padding-left: 0px; margin: 0; border: 0; list-style: none; }
.menu_2 li.menu2_llitem, .menu_2 li#active-11 {display: inline; float: left; position:
relative; text-transform: none; border-bottom: 0px solid;
    background: url("images/template/liststart.gif") transparent no-repeat; padding: 0
1px 0 8px; margin: 0 -9px 0 0; z-index: 1; }
.menu_2 li a {height: 21px; background: url("images/template/listitem.gif")
transparent right top no-repeat; color: #000000;
    padding: 0 8px 0 0; border: 0; line-height: 2em; font-size: 12px; margin: 0 10px 0
0; display: block; float: left; }
.menu_2 li:hover {}
.menu_2 li#active-11 {background: url("images/template/liststartselected.gif")
transparent no-repeat !important; }
.menu_2 li#active-11 a {background: url("images/template/listitemselected.gif")
transparent right top no-repeat; background-color: transparent; border: none; }
.menu_2 li#active-11 a:hover {}
.menu_2 li#active-12 a {background: none !important; border: none; float: none; font-
size: 11px; text-decoration: underline; }

```

```

.menu_2 li#active-l2 a:hover {;}
.menu_2 li ul {visibility: hidden; position: absolute; width: 200px; padding-left: 8px; padding-top: 5px;}
ul.menu2_level1 {background: url("images/template/menu_footer.png") transparent left bottom no-repeat; padding-bottom: 5px; left: 0em;}
.menu_2 li ul li {width: 100%; display: block; float: left; position: static; }
.menu_2 li ul li a {width: 100%; float: none; margin-bottom: -5px !important;}
.menu_2 li ul li:hover {position: relative;}
.menu_2 li ul li ul {top: 0; left: 100%;}
.menu_2 li:hover ul, .menu_2 li:hover li:hover ul, .menu_2 li:hover li:hover li:hover ul, .menu_2 li:hover li:hover li:hover li:hover ul {visibility: visible; top: 21px;}
.menu_2 li:hover li ul, .menu_2 li:hover li:hover li ul, .menu_2 li:hover li:hover li:hover li ul, .menu_2 li:hover li:hover li:hover li:hover li ul {visibility: hidden;}
.menu_2 ul li:hover ul a {;}
.menu_2 ul li ul li a:hover, .menu_2 ul li ul li:hover a {;}
.menu_2 ul li ul li a {border-style: none; margin-right: 0em;}
.menu_2 ul li ul {padding-bottom: 1px; overflow: hidden;}
.menu_2 ul li:hover ul li:hover ul a {;}
.menu_2 ul {list-style-type: none; list-style-image: none;}
.menu_2 ul li:hover a {background-color: transparent;}
.menu_2 li:hover ul {border-top: 0px solid; margin: 0; background-color: transparent; overflow: visible;}
.menu_2 ul li:hover ul li {padding: 0; margin: 0; background: none; color: #000000;}
.menu_2 ul li:hover ul li a {margin: 0; line-height: normal; width: auto; background: none;}
.menu_2 ul li ul li a:hover {background: none; }
.menu_2 li.menu2_l2item {background: none !important; }
.menu_2 li.menu2_l2item a {background: none !important; border: none; float: none; font-size: 11px; }
.menu_2 li.menu2_l2item a:hover {background: none; float: none;}
body { font: 74% "Trebuchet MS", Verdana, Lucida, Helvetica, Arial, sans-serif; font-color: Black; margin: 0; padding-left: 0; }
table { font: 99% "Trebuchet MS", Verdana, Lucida, Helvetica, Arial, sans-serif; font-color: Black; margin: 0; padding-left: 0; }
h1, h2, h3, h4, h5, h6 { color: Black; background-color: transparent; font-family: "Trebuchet MS", Verdana, Lucida, Helvetica, Arial, sans-serif; font-weight: bold; margin: 0 0 1em 0; border-bottom: 1px solid; }
h1 a, h2 a, h3 a, h4 a, h5 a, h6 a { color: Black !important; text-decoration: none; }
h1 { font-size: 150%; border-bottom: 2px solid; }
h2 { font-size: 140%; }
h3 { font-size: 125%; }
h4 { font-size: 110%; }
h5 { font-size: 100%; }
h6 { font-size: 90%; }
p { margin: 0 0 1em 0; line-height: 1.5em; }
ul { line-height: 1.5em; margin: 0.5em 0 0 1.5em; padding: 0; list-style-type: square; }
ol { line-height: 1.5em; margin: 0.5em 0 0 2.5em; padding: 0; list-style-image: none; }
li { margin-bottom: 0.5em; }
dt { font-weight: bold; }
dd { line-height: 1.5em; margin-bottom: 1em; }

/** Settings below this line are from Plone-site **/

```

```

fieldset { border: 1px solid; margin: 1em 0em 1em 0em; padding: 0em 1em 1em 1em; line-
height: 1.5em; width: auto; }
legend { background: White; padding: 0.5em; font-size: 90%; }
form { border: none; margin: 0; }
label { font-weight: bold; }
button { font-family: Verdana, Lucida, Helvetica, Arial, sans-serif; visibility:
visible; border: 1px solid; color: Black; vertical-align: middle; padding: 1px;
cursor: pointer; font-size: 90%; text-transform: lowercase; }
select { border: 1px solid; color: Black; background-color: White; vertical-align:
top; }
optgroup { font-style: normal; font-weight: bold; color: #999; padding-left: 0.25em; }
option { color: black; }
optgroup > option { padding: 0 0.25em 0 1em; }
abbr, acronym, .explain { border-bottom: 1px dotted Black; color: Black; background-
color: transparent; cursor: help; }
q { font-family: Baskerville, Georgia, serif; font-style: italic; font-size: 120%; }
blockquote { padding-left: 0.5em; margin-left: 0; border-left: 4px solid; color:
#000000; }
code, tt { font-family: Monaco, "Courier New", Courier, monospace; font-size: 120%;
color: Black; padding: 0 0.1em; }
pre { font-family: Monaco, "Courier New", Courier, monospace; font-size: 100%;
padding: 1em; border: 1px solid; color: Black; overflow: auto; }
ins { color: green; text-decoration: none; }
del { color: red; text-decoration: line-through; }
.visualClear { display: block; clear: both; }
.netscape4 { display: none; }

```

## style-mercator.css

### Mercator-Research –site-specific stylesheet

2 other sites are identical except they don't have language flags and plug-in (calendar & multi-content) settings, and of course the values for the colors are different

```

/* positioning of the stuff inside the header */
#header { background: url(images/template/gradient75.png) top right; background-
repeat: repeat-y; }
#merclogo { float: left; margin-top: 5px; margin-left: 10px; }
#part-of-fa { float: right; position: relative; top: 20px; margin-right: 20px; }

/* colors */
h1, h2, h3, h4, h5, h6 { border-color: #6aa94e; }
a:link { color: #6aa94e; }
a:hover { color: #34C014; text-decoration: underline; }
a:visited { ; }
hr { color: #6aa94e; }
blockquote { border-color: #6aa94e; }
#footer { clear: both; }
ul { list-style-image: url("images/template/bullet.gif"); padding-left: 5px; }

```

```

#main_content .csc-mailform { border-color: #6aa94e; }

/* Boxes */
#boxes_left DIV.csc-header, #boxes_right DIV.csc-header
{ border-bottom: 1px #548B54 solid; background:
url("images/template/portlet_header.gif") no-repeat top; }
.box_footer
{ background: url("images/template/portlet_footer_inv_new.png") no-repeat bottom; }

/* Menu colors */
.menu_2 {border-color: #6aa94e; }
.menu_2 a:hover {color: #34C014; }
.menu_2 li#active-11 a {color: #34C014; }
.menu_2 li#active-11 a:hover {color: #34C014; }
.menu_2 li:hover ul {border-color: #6aa94e; }
.menu_2 li a {background: url("images/template/listitem.gif") transparent right top
no-repeat; }
.menu_2 li.menu2_11item, .menu_2 li#active-11 {background:
url("images/template/liststart.gif") transparent no-repeat; }
ul.menu2_level1 {background: url("images/template/menu_footer_new.png") transparent
left bottom no-repeat; }

/*===CALENDAR===*/ // MERCATOR-RESEARH ONLY
#main_content .tx-cal-controller .month-large td {
border-style: groove;
border-color: #6aa94e;
border-width: 2px;
}

/*===THE HOMEPAGE MULTICONTENT PLUGIN SETTINGS===*/ // MERCATOR-RESEARCH ONLY
.tx-jfmulticontent-pil {
border-style: solid;
border-color: #6aa94e;
border-width: 2px;
border-top: none; /*border disabled*/
border-bottom: none; /*border disabled*/
border-left: none; /*border disabled*/
border-right: none; /*border disabled*/
padding-top: 0px;
padding-bottom: 0px;
margin-left: auto;
margin-right: auto;
}
.tx-jfmulticontent-pil DIV.csc-default { margin-left: 0px; margin-right: 0px; }
.tx-jfmulticontent-pil DIV.csc-header{ visibility: hidden; text-align: right; font-
size: 0%;}
.tx-jfmulticontent-pil a { color: #6aa94e !important; }
.tx-jfmulticontent-pil a:hover { color: #34C014 !important; text-decoration: underline
!important; }
.ui-widget-header { border-color: #6aa94e !important; background: url(image/.png) 50%
50% repeat-x #6aa94e !important; }

```

## 5 JavaScripts

(for font resizing and saving the chosen size to cookie locally)

### resizeFont.js

```
function resizeFont(multiply) {
    var sz = document.body.style.fontSize;
    if (sz == '') sz = 74; //default font size
    if (sz < '74') sz = 74; //minimum
    if (sz > '110') sz = 110; //maximum

    var size = parseFloat(sz) + (multiply * 6) + "%";
    document.body.style.fontSize = size;
    SetCookie("fs",size,30); //store the size value in the cookie for 30 days
}
```

### setFont.js

```
function setFont() {
    var sz = ReadCookie('fs');
    if (sz != null && sz != '') {
        document.body.style.fontSize= sz;
    }
}
```

### setCookie.js

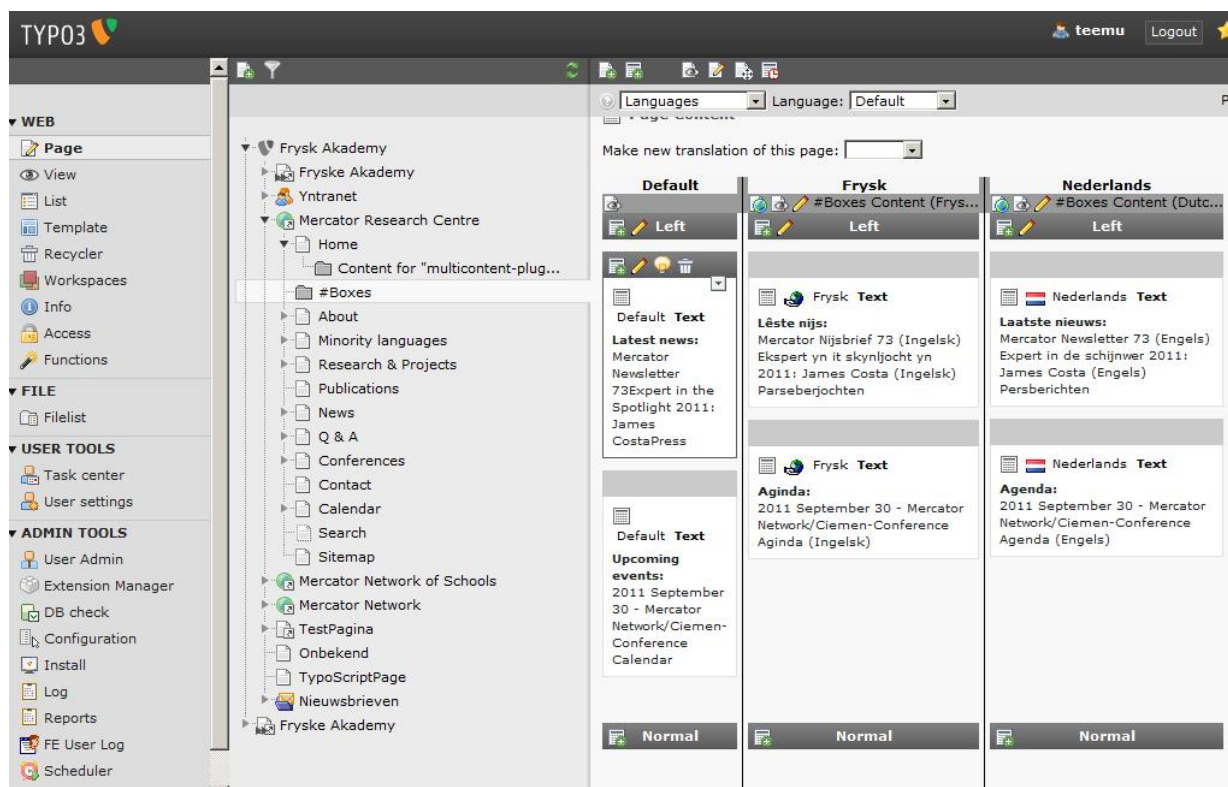
```
function SetCookie(name,value,days) {
    var now= new Date();
    var expDate = new Date();
    if (days==null || days==0) days=1;
    expDate.setTime(now.getTime() + 3600000*24*days); //create date after no of
    "days" from now
    //create cookie with name, value and expire date
    document.cookie=name+"="+escape(value)+"; path=/;
    expires="+expDate.toUTCString();
}
```

### ReadCookie.js

```
function ReadCookie(name) {
    name = name.replace(/([.\|\+\\-*/?:!|^$}{()\\[\]])/g, '\\$1');
    var re = new RegExp('; ' + name + '=(\\^\\s;]*)');
    var sMatch = (' ' + document.cookie).match(re);
    if (name && sMatch) return unescape(sMatch[1]);
    return '';
}
```

## 6 Manuals

### *How to add content to “boxes” and the limitations*



The current template settings set following restrictions in order the boxes to display correctly on page:

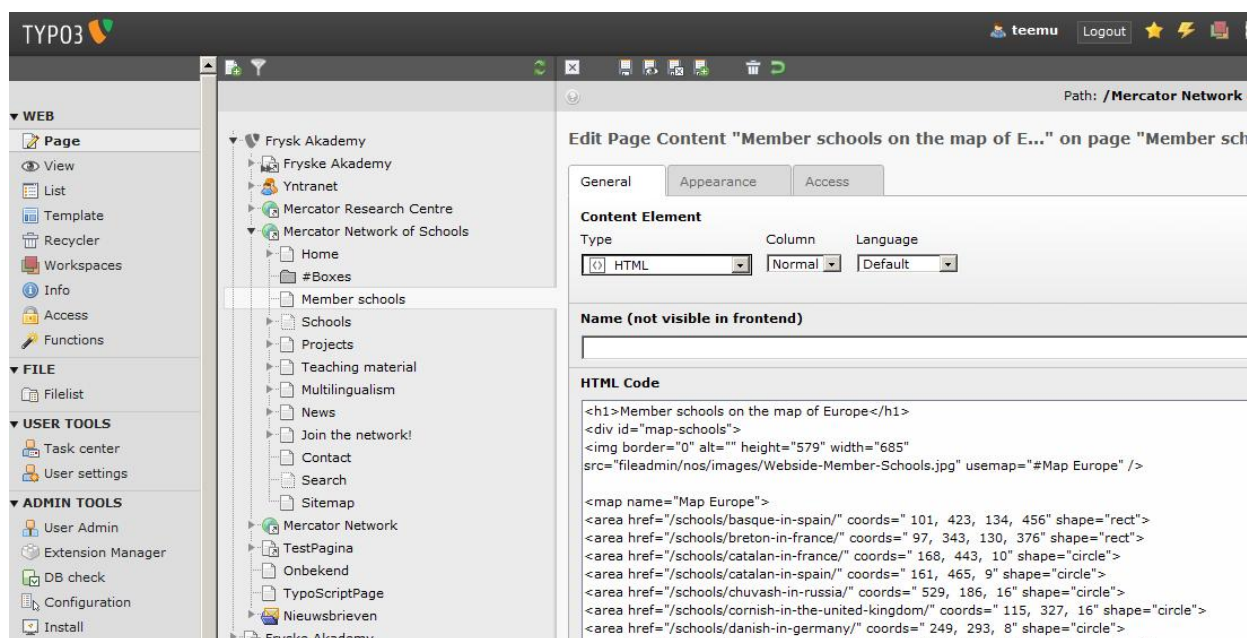
- Every site has a special folder in the root directory named “#Boxes” where the content for the boxes are stored (see the picture above). These could be called “site-wide boxes”. Content inserted in this folder is displayed on every page of that site (as defined in the TypoScript-code).
- Any content added to Left and Right columns on *ANY OTHER* page is displayed in the output on these *specific* pages only, and they are always placed on top of the “site-wide boxes” explained above.
- The implementation allows any type of content to be inserted in the “boxes”. The image used to create the footer and side-borders is 1000 pixels high so that is the vertical limit.

-The title of the box must be typed into the “header”-section. Otherwise the box won’t have header that is required in order to boxes to display graphically correctly.

-The boxes’ width is set to 154px which was the original width as well. This can of course be changed, but the main-container in which the “boxes” and “main content” <DIV>s are is 100% full, so even 1px increase of any DIV or margin causes the right-side DIV to break the visual design.

## *Network of Schools site’s “clickable map”-manual*

The HTML-code for the new objects has to be manually inserted.



Make a copy of one of the existing rows that starts with <area href=...

The page-path to the new “dot” can be seen and copy & pasted from the public web-browser after that specific page has first been created, but if created as the existing ones, the path should be /schools/school-name/ where “school-name” is replaced with the actual name,

New objects’ coordinate data should be found using a separate graphics application, and then added as was the previous procedure (separate manual exists for this).

The new image with a new dot should be overwritten to the existing image file, which is placed in the images –directory under “nos” as “Webside-Member-Schools.jpg”.

### *How to add/remove/modify Calendar events*

The calendar is implemented exactly the same way as the Fryske Akademy -site's equivalent.

In the TYPO3 backend under the Calendar-page there is a special folder called "Calendar events". The folder should be opened in the "List"-view. New events can be added as with the main site and they will appear in the calendar.

### *How to add link to upcoming event to the box*

To add the event link to the "box", this has to be done manually as the calendar does not give option to show them in the desired way.

On the other hand though, this gives the possibility to write the title of the link as is most fitting, as well as allows the usage of images as well.

In order to add a link to the "Event"-page inside the box, the event must first be created and then opened in any web browser. Copy that link to clipboard and paste as a link manually. As this is going to be external link, make sure in the link properties the "target" is set as "\_self" in order the link not to open in new browser window or tab.

### *How to use Multiple Content –plugin extension*

(Mercator Research homepage only)

Mercator Research site's homepage includes lots of content. To make this more attractive and modern-looking, but at the same keeping it simple, an extension called "jfmulticontent" was installed to allow many pages to be shown on one page under different kinds of tabs.

This plugin has its own configuration.

A guide to the main things:

- There is a folder called "content for multicontent plugin" under the Home-page of the Mercator Research -site. This is where the pages to be shown on the front-page are stored. Handling the translated pages goes as normal. Be sure to add the title of the pages to the appropriate "Header"-part (not inside the main content) as the plugin uses these to display the titles.



- A creation of new pages in this folder does not add them automatically to the multicontent display however. The pages need to be added separately under the configuration of the plugin on the "Home"-page. Only the Default (English)-page of the plugin settings is supposed to be touched, the translations are shown automatically based on the active sys\_id which TYPO3 handles internally.
- Under the plugin-settings the content items can be added, removed and their order of display can also be defined.
- All the other settings should be self-explanatory I hope.

### *Visitor information –page needs to be updated only on one page*

As the visitor information on all three sites is identical, in case of an update, the identical part needs to be updated only on one page.

The master "visit us" –page is in the Mercator Research's "Contact"-folder.

In case the page is wanted to be changed in the future for the specific site, the "Mount point"-entries can easily be deleted/changed.

### *Good to know about deleted page content items*

Be aware that even you delete a content item in TYPO3 (at least in the current installed version 4.5.2), it remains in the database and sometimes (depends on the type of the content item) continues to be shown publically if it was enabled at the time of deletion.

When page content item is deleted the object is simply just moved to the Recycler. In order to delete the content completely, the item must be deleted permanently from the Recycler.

### *In case the main menu items are modified*

If the main menu names are modified in any way, especially if more items are added or removed, or their length is increased or decreased significantly, the site-specific CSS-stylesheet might require small adjustment in order the menu bar to be properly positioned.

Take note that on the multilingual Mercator Research Centre –site, the menu has different width in different languages, so adjust according to the longest.

In the TYPO3 backend, open "File->Filelist" –view. In the "/fileadmin" –directory, there is a directory for each site ("mercator", "network" and "nos"), open the directory that you need to adjust. Inside the folder there are many files, but the one we are interested in is the site-specific CSS-stylesheet, named respectively: "style-mercator.css", "style-network.css" or "style-nos.css". Click the "Edit"-button next to the filename.

Find a row that starts with string: `#main_menu {margin-left: XXXpx;} /* MENU'S DISTANCE FROM LEFT!! */`

This is the setting that needs to be adjusted.

Enter a new integer and save the file. If a new item was added or some title was made longer, the integer must be smaller. (padding adjusts the menu's distance from the left edge of the website)

Clear the caches, and in other browser window or tab, open site's page inside the browser and see if you got it right or not. Just keep changing the number until you get it right.

### *Main numerical design-limitations collected in one from all the above*

A width of the main content container is 640 pixels (15 pixels of marginal on left and right side).

Inside the main content container, the HTML-content items [h1, h2, h3, h4, h5, h6, p, ul] are given 15 pixels of padding on the left side to make nicer visual appearance.

A width of the boxes is 154 pixels. This is especially important to know when designing the photos to be inserted inside the boxes.